# MODELLING AND SIMULATION 2009

## THE EUROPEAN SIMULATION

## AND

## MODELLING CONFERENCE

## 2009

# ESM®'2009

EDITED BY

Marwan Al-Akaidi

OCTOBER 26-28, 2009

LEICESTER, UNITED KINGDOM

A Publication of EUROSIS-ETI

# The European Simulation and Modelling Conference 2009

LEICESTER, UNITED KINGDOM

OCTOBER 26-28, 2009

Organised by

ETI- The European Technology Institute

Sponsored by

EUROSIS, The European Simulation Society

Co-Sponsored by

**Ghent University**

**De Montfort University**

**and**

**IEEE UKRI-SPC**

Hosted by

Holiday Inn

Leicester, United Kingdom

# INTERNATIONAL PROGRAMME COMMITTEE

**High Performance Large Scale and Hybrid Computing**
**Track Chair:** Beniamino di Martino, Second University of Naples, Italy
Jan Broeckhove, University of Antwerp, Antwerp, Belgium
Giancarlo Fortino, Universita della Calabria, Rende, Italy
Jari Porras, Lappeenranta University of Technology, Finland
Simon See, Sun Microsystems Inc, Singapore
Pierre Siron, ONERA, Toulouse, France
Behrouz Zarei, Sharif University of Technology, Tehran, Iran

**Parallel Processing with Games Machines**
Sofie Van Volsem, Ghent University, Ghent, Belgium

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

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

**Analytical and Numerical Modelling Techniques**
Ana M. Camacho, UNED, Madrid, Spain
Tom Dhaene, Ghent University, Ghent, Belgium
Clemens Heitzinger, University of Vienna, Vienna, Austria
Panajotis Katsaros, Aristotle University, Thessaloniki, Greece
Eva M. Rubio, UNED, Madrid, Spain

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

**Agent Based Simulation**
Eric Gouarderes, UPPA, Pau, France
Frederic Guinand, Universite du Havre, France
Zisheng Huang, Vrije Universiteit Amsterdam, The Netherlands
Jean-Luc Koning, INPG-ESISAR-LEIBNIZ, Grenoble, France
Peter Lawrence, Australian Catholic University, Melbourne, Australia
Ioan Alfred Letia, TU Cluj Napoca, Romania
Paulo Novais, Universidade do Minho, Braga, Portugal
Jan-Torsten Milde, FH Fulda, Germany
Isabel Praca, Ist. Superior do Porto, Portugal
Marco Remondino, University of Turin, Italy
Agosthino Rosa, Technical University Lisbon, Portugal

# INTERNATIONAL PROGRAMME COMMITTEE

**Simulation with Petri Nets**
Pascal Berruet, Universite Bretagne Sud, Lorient, France
Carmen Bobeanu, Ghent University, Belgium
Mauro Iacono, University of Naples II, Italy
Juan de Lara, Univ. Autonoma de Madrid, Spain
Hammid Demmou, LAAS CNRS, Toulouse, France
Olivier Grunder, UTBM, Belfort, France
Guenter Hommel, TU Berlin, Germany
Stefano Marrone, Seconda Universita degli Studi di Napoli, Naples, Italy
Alexandre Nketsa, LAAS-CNRS, Toulouse, France
Mario Paludetto, LAAS-CNRS, Toulouse, France
Jean-Claude Pascal, LAAS-CNRS, Toulouse, France
Ivo Vondrak, Technical University of Ostrava, Czech Republic

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

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

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

# EUROPEAN SIMULATION AND MODELLING CONFERENCE 2009

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

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

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

EUROSIS-ETI Publication

**ISBN: 978-90-77381-52-6**
**EAN: 9789077381526**

# Preface

Dear participants

It is my pleasure to welcome you to the 2009 European Simulation and Modelling Conference (ESM® 2009), the international European conference on the state of the art of modelling and simulation, which this year is being held at the Holiday Inn, in the city of Leicester, United Kingdom in cooperation with the de Montfort University.

Even though we live in harsh economic times with declining numbers in participation, this year's event still has managed to attract some 65 high quality papers from 21 different countries spanning 4 continents, out of 86 papers submitted.

Further to the selected scientific presentations, EUROSIS and I are grateful to Professor Adrian Hopgood of de Montfort University for giving this year's keynote speech entitled: "Hybrid Systems, the Future of Artificial Intelligence" and to our invited speakers; Ken Kahn from Oxford University with his talk on "The Modelling4All Project: A web-based modelling tool embedded in Web 2.0" and Simon Scarle from Warwick University with his talk on "Putting a Heart into a Box: GPGPU simulation of a Cardiac Model on the XBox 360".

I wish to thank all those, who have contributed their time and effort in organizing this meeting. This goes out to the International Program Committee members who took care of the reviewing process. They have done a great job in arranging a strong technical program, which covers a variety of speciality areas covering present day methodological simulation research.

Recognition for this conference must go also to Philippe Geril, the EUROSIS coordinator, who was the main force responsible for the organisation of the meeting.

Furthermore, I would like to thank the Creative Technology Studios at de Montfort University, for accepting to have the conference participants visit the BBC research studios at the aforementioned site.

Finally, I would like to wish you a pleasant stay in Leicester and a successful conference meeting

Professor Dr Marwan Al-Akaidi
ESM'2009 General Conference Chair
EUROSIS – M. East Chair
School of Engineering & Technology,
De Montfort University,
Leicester, LE 1 9BH, UK.
Email: mma@dmu.ac.uk

# CONTENTS

## MODELLING ENVIRONMENTS

## MODULAR SIMULATION AND DESIGN

## MODEL VERIFICATION, VALIDATION AND EVALUATION

# CONTENTS

## SIMULATION TOOLS

## LOGISTICS AND WAREHOUSING TOOLS

# CONTENTS

## MANUFACTURING MANAGEMENT TOOLS

## ENGINEERING SIMULATION

## ENERGY SIMULATION

## HEALTH SERVICE MANAGEMENT

# CONTENTS

## DATA SIMULATION AND STORAGE

## DECISION SUPPORT SYSTEMS

## SIMULATION AND AI

## BEHAVIOURAL MODELLING

## AGENT BASED SIMULATION

# CONTENTS

## WATER MANAGEMENT SYSTEMS

## FLUID FLOW SIMULATION

## PLUME SIMULATION

# SCIENTIFIC PROGRAMME

# MODELLING ENVIRONMENTS

# OPTFERM – A COMPUTATIONAL PLATFORM FOR THE OPTIMIZATION OF FERMENTATION PROCESSES

Orlando Rocha[1,2,3], Paulo Maia[1,2], Isabel Rocha[1,3], Miguel Rocha[2]
[1]IBB – Institute for Biotechnology and Bioengineering / Centre for Biological Engineering
[2]CCTC – Computer Science and Technology Center / Dep. Informatics - Universidade do Minho
Campus de Gualtar, 4710-057 Braga, Portugal
E-mails: {orocha,pmaia,irocha}@deb.uminho.pt, mrocha@di.uminho.pt
[3]Biotempo, Lda., Avepark – Zona Industrial Gandra, Apartado 4152, 4806–909, Caldas Taipas, Portugal

**KEYWORDS**
Fermentation processes, open-source software, process simulation and optimization, Evolutionary Computation, Differential Evolution

## ABSTRACT

We present *OptFerm*, a computational platform for the simulation and optimization of fermentation processes. The aim of this project is to offer a platform-independent, user-friendly, open-source and extensible environment for Bioengineering process optimization that can be used to increase productivity. This tool is focused in optimizing a feeding trajectory to be fed into a fed-batch bioreactor and to calculate the best concentration of nutrients to initiate the fermentation. Also, a module for the estimation of kinetic and yield parameters has been developed, allowing the use of experimental data obtained from batch or fed-batch fermentations to reach the best possible model setup.

The software was built using a component-based modular development methodology, using Java as the programming language. *AIBench,* a Model-View-Control based application framework was used as the basis to implement the different data objects and operations, as well as their graphical user interfaces. Also, this allows the tool to be easily extended with new modules, currently being developed.

## INTRODUCTION

Nowadays, several products such as antibiotics, proteins, amino-acids and other chemicals are produced using fermentation processes. Due to the rise of petroleum prices and the incentive to replace petroleum derivatives by “green products”, many traditional processes have been replaced by new biotechnological ones. Consequently, an effort to improve biotechnological techniques has been verified. Recombinant DNA applications were conceived to produce new microorganisms, while several computational tools have been designed and implemented for modelling and simulation of metabolic pathways of the cell (Pettinen et al. 2005). All share a common purpose: to increase the production yield and get a higher purity of the final product.

To optimize the productivity of a biological process, in the majority of the cases, two different steps have to be addressed: firstly, a selection and genetic improvement of the organism strain is accomplished; in a second step, the best conditions of the fermentation process are identified, such as the initial nutrient concentrations, operating modes, feeding profiles for fermentations, temperature and pH. In industry, the second step is mostly done experimentally using trial-and-error heuristics (Kawohl et al. 2007). Although there are several tools to study, simulate and optimize cellular pathways, there is still a clear lack of tools to perform the optimization of fermentation processes.

Fermentation processes are affected by biochemical and chemical phenomena such as the chemical interactions between components, concentrations of substrates, products and biomass, and environmental conditions like temperature, pH and dissolved oxygen concentration (Tzoneva 2006; Zhang 2008). The complex dynamic behavior and the unpredictable effects of these factors increase the difficulty of establishing accurate models to describe the real systems (Benjamin et al. 2008). However, new methods to control, predict and optimize bioprocesses have been proposed.

The OptFerm platform was developed using the Java programming language, with the aim of being a user-friendly, extensible and platform-independent tool. It was designed to allow the user to evaluate and compare several different methods for the tasks of simulation, optimization and parameter estimation, in the context of fermentation processes. The aim is to allow users to improve process productivity, achieving better results in reduced times.

The available optimization algorithms in this tool were developed and validated in previous work by the authors, namely Evolutionary Algorithms (Rocha et al. 2004, 2007; Mendes et al 2006, 2008), Differential Evolution (Mendes et al. 2006, 2008) and Simulated Annealing (Rocha et al. 2007). Any of these algorithms can be used in feed optimization or parameter estimation. Metaheuristic optimization approaches are used, since the underlying problems are typically quite complex. OptFerm is available in the following website: http://darwin.di.uminho.pt/optferm.

## MAJOR FEATURES

The main aim of the OptFerm software is to provide specific computational tools for the simulation and optimization of fermentation processes. The tools should enable its users to use several methods and parameter configurations, thus saving time in performing expensive wet experiments.

### Fermentation models

The basis for all operations available within OptFerm are the models of the fermentation processes. The internal representation of a model is based on ordinary differential equations (ODEs). In OptFerm, model information can be divided in two main entities, a *Process* and a *Function:*

- *Process* – contains information on the state variables such as names, initial values and upper and lower limits, and the objective function for optimization purposes.
- *Function* – keeps the kinetic parameters (names, values and limits), kinetic reactions and the ODEs that describe the current problem dynamics.

The kinetic reactions and the ODEs are defined separately, allowing any type of kinetic equations to be defined for a given set of ODEs. The user can apply constraints to limit or impose a condition when a value of a state variable or kinetic reaction is exceeded. The kinetics functions can be implemented using any of the control flow statements in Java, demanding some knowledge of the programming language, but allowing a greater flexibility.

The dynamical model describing the state variables behavior along time is described by a set of ODEs (see the case study). There are only two restrictions in the definition of the model: it is necessary to associate a substrate feeding rate parameter and a dilution rate factor has to be associated to all differential equations, with the exception of the equation describing volume/ weight variations.

Currently, the ODEs and kinetics have to be written in the Java language. The definition of a new model requires the implementation of two classes: one for the Process and the other for the Function. The structure of these classes is always the same, since they are based on a common interface. After the compilation of a model, the different data values associated with it are considered as default data and cannot be modified. Nevertheless, new instances can be created with different sets of values for different parameters. Indeed, when a Project is created, new sets of initial values for state variables, model parameters and feeding profiles can be defined and kept for future use.

**Simulation**

Regarding the process simulation, the user has the ability to test various combinations of the initial values for state variables, parameters and experimental or hypothetical feeding trajectories along time. Furthermore, it is possible to perform simulations with feed trajectories obtained from optimization. Likewise, after executing the estimation of model parameters, the results are immediately accessible and can be used to perform a simulation. The simulation results can also be compared with experimental data. The results are displayed via graphs, where each state variable or kinetic rate can be visualized separately. These figures can be exported as JPEG files. Simulations are performed by running a numerical integration process, using a linearly implicit-explicit Runge-Kutta scheme or a constant Runge-Kutta scheme, included in OdeToJava (Ascher et al. 1997).

**Optimization**

Three types of operations can be performed: the optimization of a simple feeding trajectory, of the feeding trajectory plus initial conditions or of the feeding trajectory plus final time (Rocha et al, 2004). In the first case, the ideal amount of substrate to be fed into the reactor per time unit along time is determined; the second scenario allows determining the best initial concentrations for each selected state variable, while in the third case the optimal duration of the fermentation is also provided.

The minimum and maximum pump limits can be defined by the user and these values are used as constraints on the optimization operations. Some preferences related with the algorithms can be modified by the user, such as the number of iterations, the population size, the discretization step and an interpolation factor. This factor is used to reduce the solution size, so that feeding values are defined only at certain equally spaced points. A report on the optimization operations performed can be generated, describing the conditions that were used and the results obtained.

**Parameter estimation**

To perform the estimation of parameters, a simple GUI is available, where the various estimation options are easily understandable. It is possible to fix certain parameters or to assert that certain state variables should be ignored during the estimation (this is important because if a state variable has null values over time, the objective function is affected, causing a numerical error). The results are presented in graphs or tables and both can be saved to files. As with feed optimization, a report can be generated. The fitting is performed by minimizing a total cost function that represents the adjustment between experimental and simulated data:

$$Total\ Cost = \sum_{i=1}^{n}\left(\frac{1}{N_p}\sum_{j=1}^{p}\left(\frac{\xi_{sim,ij}-\xi_{\exp,ij}}{\bar{\xi}_{\exp,ij}}\right)^2\right) \quad (1)$$

where $\xi_{sim,ij}$ represents the simulated data and $\xi_{\exp,ij}$ the experimental data for the state variable ξ (n is the number of state variables) for every point (p is the total of data points). The difference is divided by an average value $\bar{\xi}_{\exp,ij}$ with the purpose of giving the same importance to all state variables.

**IMPLEMENTATION ISSUES**

OptFerm is built in a modular way, using a component-based approach to software development. AIBench, a general purpose Java application framework for scientific software development, was used to manage the data objects and execute the operations, also making the linkage with the graphical interface. All information related with AIBench can be found in http://www.aibench.org/.

AIBench is a MVC (Model-View-Control) based Java application developed by the University of Vigo, with the collaboration of the authors. It uses a plug-in engine, which provides the capability to load or unload operations, allowing to create applications based in software modules. All applications developed with AIBench are structured through two main concepts: *datatypes*, defining data structures used in the application and *operations* describing functions receiving input objects and creating output objects. To implement OptFerm, it was necessary to define the corresponding datatypes and operations. A general schema of OptFerm structure is shown in Figure 1.

A *datatype* is a Java class that specifies the internal representation of an object, in which simple data or complex data (other datatypes) are incorporated. It may be considered as a container. They can be used or created during the various operations.

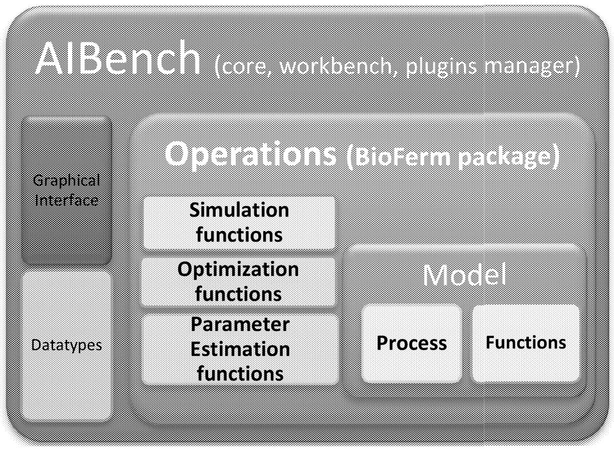

AIBench (core, workbench, plugins manager)
Graphical Interface
Datatypes
Operations (BioFerm package)
Simulation functions
Optimization functions
Parameter Estimation functions
Model
Process
Functions

Figure 1: The general internal structure of *OptFerm*

In OptFerm, the datatypes were structured as (Figure 2):

- **Project** – it is the basic datatype; when a Project is created, each of the objects shown in Figure 2 is instantiated. A Project is directly related to a model (it has to contain one Model object and cannot contain more than a single one). A project has a list of Simulation, Optimization and Parameter Estimation results. These lists are extended, during the execution of each operation.
- **Model** – Within each Model there are four different datatypes, as shown in Figure 2, namely: State Variables, Kinetic Parameters, Feed Data and Experimental Data. These datatypes are of type List, in which a new set of initial values for the state variables, parameters, feed profiles or an experimental dataset can be added to the list. Consequently, different combinations of state variables, feed profiles and kinetic parameters can be used in the simulation, optimization and parameter estimation operations, without the need to change the internal structure of the model.
- **Simulation, Optimization and Estimation Results** – these are datatypes of type List. After the execution of each of these operations, a new object is created containing the results. The conditions that were used in these operations are saved, such as state variables, model parameters, feed profiles and experimental data sets.

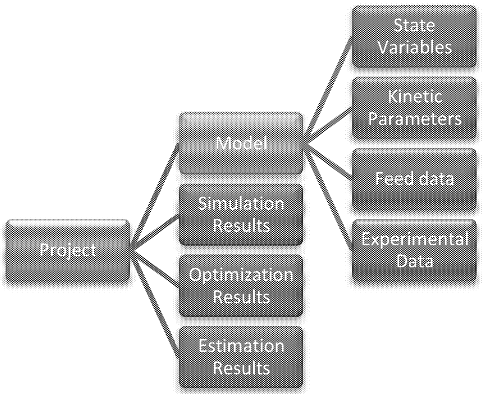


Figure 2: Structure of the Datatypes within a Project

All datatypes are organized in a *Clipboard* and presented to the user as a tree. The data contained inside the datatypes can be accessed through viewers, graphical interfaces where data is presented in tables, graphics or other suitable means.

In terms of the source code organization, a main library gathers the various packages with the simulation, optimization and estimation functions and a description of the models. A module containing specific optimization routines were created for feed optimization and related tasks. This module uses JECoLi (Java Evolutionary Computation Library; http://darwin.di.uminho.pt/jecoli) that contains generic optimization routines based on metaheuristic search algorithms. Some adaptations had to be made to adapt these algorithms to support feed optimizations, as explained in Rocha et al (2004) and Mendes et al (2006). Three algorithms belonging to the main groups of Evolutionary Algorithms, Differential Evolution and Simulated Annealing are used to perform optimizations.

A package for kinetic parameter estimation was developed, using the same optimization routines. Some modifications were made to enable the user to perform estimations without needing to modify the internal structure of the models. Functions to import/ export data were also implemented.

## CASE STUDY

The case study is related to production of ethanol by *Saccharomyces cerevisiae* , described by Chen and Huang (1990). The purpose is to explain in a descriptive way the most important features of OptFerm and not to make any study of the used model. Due to space constraints only Simulation and Optimization operations are considered.

The model represents a fed-batch bioreactor system and encompasses the following equations (Chen e Hwang 1990):

$$\frac{dx_1}{dt} = g_1 x_1 - u\frac{x_1}{x_4} \qquad (2)$$

$$\frac{dx_2}{dt} = -10 g_1 x_1 + u\frac{150 - x_2}{x_4} \qquad (3)$$

$$\frac{dx_3}{dt} = g_2 x_1 - u\frac{x_3}{x_4} \qquad (4)$$

$$\frac{dx_4}{dt} = u \qquad (5)$$

where $x_1$, $x_2$ and $x_3$ are the cell mass, substrate and ethanol concentrations (g/L), $x_4$ the volume of the reactor (L) and $u$ the feeding rate (L/h). Kinetic variables are given by:

$$g_1 = \frac{0.408}{1+\frac{x_3}{16}} \frac{x_2}{0.22 + x_2} \qquad (6)$$

$$g_2 = \frac{1}{1+\frac{x_3}{71.5}} \frac{x_2}{0.44 + x_2} \qquad (7)$$

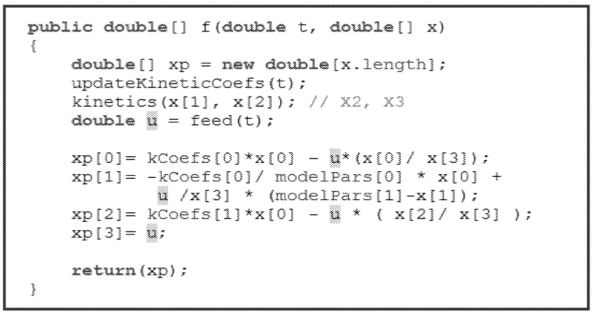
The objective function was set to obtain a maximum of $x_3$ when the maximum of reactor capacity ($x_4$) is reached:

$$prod = x_3(T_f)x_4(T_f) \qquad (8)$$

where $T_f$ is the final time.

### Model edition

The first step was to define the *Process* Java class and the *Function* Java class. The ODEs (equations 2 to 5) are converted into the equations presented in Figure 3. This represents a function that receives the present time value and an array of state variables calculated in the previous iteration. Next, it calls the *updateKineticCoefs(t)* method to calculate new values for the kinetic variables at time *t*, and then it calculates and returns an array containing the new values for the state variables at time *t*.



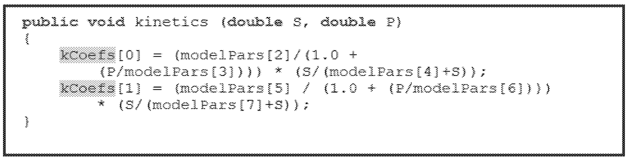
```
public double[] f(double t, double[] x)
{
    double[] xp = new double[x.length];
    updateKineticCoefs(t);
    kinetics(x[1], x[2]); // X2, X3
    double u = feed(t);

    xp[0]= kCoefs[0]*x[0] - u*(x[0]/ x[3]);
    xp[1]= -kCoefs[0]/ modelPars[0] * x[0] +
           u /x[3] * (modelPars[1]-x[1]);
    xp[2]= kCoefs[1]*x[0] - u * ( x[2]/ x[3] );
    xp[3]= u;

    return(xp);
}
```

Figure 3: How ODEs are defined in the Function Java class

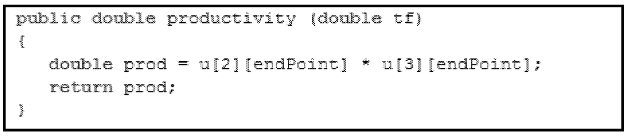
The kinetic equations 6 and 7 have to be converted to the Java language as shown in Figure 4 . The variables $g_1$ and $g_2$ at each iteration are saved in an array *kCoefs*, and these values are used later in the ODEs. The *modelPars* are the kinetic parameters defined in the *Function* java class as well.



```
public void kinetics (double S, double P)
{
    kCoefs[0] = (modelPars[2]/(1.0 +
        (P/modelPars[3]))) * (S/(modelPars[4]+S));
    kCoefs[1] = (modelPars[5] / (1.0 + (P/modelPars[6])))
        * (S/(modelPars[7]+S));
}
```

Figure 4: How kinetic variables are defined in Java

An objective function must be defined, describing the purpose of the optimization. The aim was to obtain the maximum of ethanol ($x_3$) and equation 8 was used, being defined in the *Process* class as the *productivity* method:



```
public double productivity (double tf)
{
   double prod = u[2][endPoint] * u[3][endPoint];
   return prod;
}
```

Figure 5: The objective function in the Java language

### OptFerm ClipBoard

After defining the process and function classes, these are compiled and are ready to use in OptFerm. A new project is created and all initial Datatypes are displayed (Figure 6). They are presented as a tree structure, and the data contained can be viewed by simply clicking over the datatypes. All functionalities are displayed in the menus.

Different sets of initial values for state variables, kinetic parameters, feeding profiles and experimental data can be created and added to the clipboard (Figure 7). Data can also be removed from the clipboard. The internal data of these new sets can be modified when necessary, and the user can save or load previously saved data.

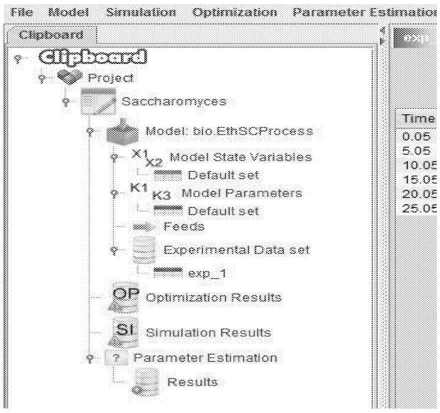


Figure 6: Example of OptFerm Clipboard

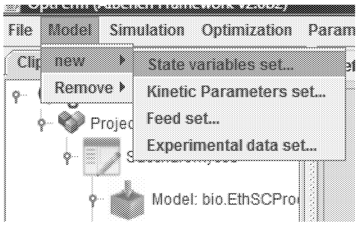
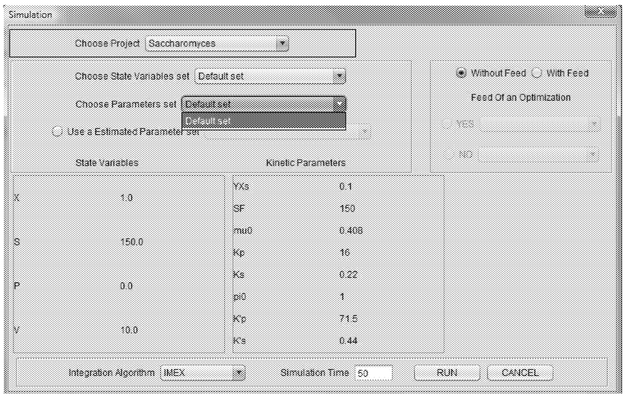


Figure 7: Menus and sub-menus of the OptFerm toolbox: example on how a new set of state variables can be created

### Simulation

An interface is presented to the user with all options to perform simulations (Figure 8). A *Project*, the initial values of state variables and kinetic parameters have to be selected. It is possible to select between feeding profiles that had been defined by the user and the ones resulting from optimization procedures. After performing a parameter estimation, the model parameters are also available to be used.



Simulation

Choose Project: Saccharomyces

Choose State Variables set: Default set

Choose Parameters set: Default set

○ Use a Estimated Parameter set

● Without Feed ○ With Feed

Feed Of an Optimization

| State Variables | | Kinetic Parameters | |
|---|---|---|---|
| X | 1.0 | YXs | 0.1 |
| | | SF | 150 |
| | | mu0 | 0.408 |
| S | 150.0 | Kp | 16 |
| | | Ks | 0.22 |
| P | 0.0 | pi0 | 1 |
| | | K'p | 71.5 |
| V | 10.0 | K's | 0.44 |

Integration Algorithm: IMEX

Simulation Time: 50

Figure 8: The graphical interface to perform simulations

After performing a simulation, the results are displayed in a graph (Figure 9). The state variables or kinetic rates can be visualized. The right panel displays the parameters used.

### Optimization

To perform an optimization, a panel is presented (Figure 10). On this panel, several options can be selected and the available sets of initial values for state variables and model parameters are displayed.

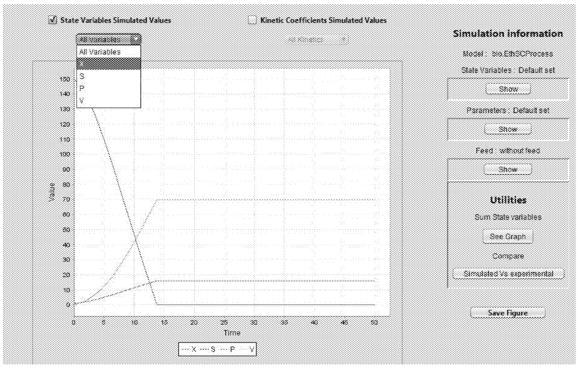
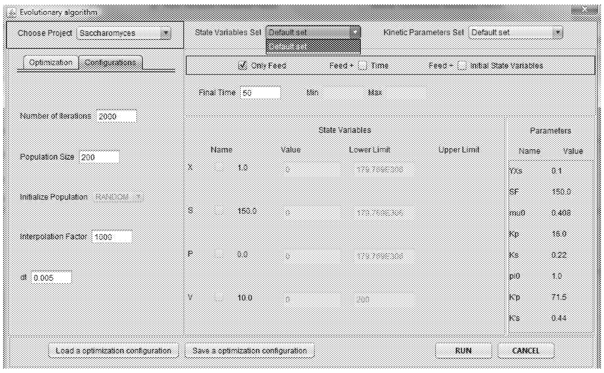


Figure 9: How simulation results are presented to the user


Evolutionary algorithm
Optimization
Configurations
☑ Only Feed
Feed + □ Time
Feed + □ Initial State Variables
Min
Max
Load a optimization configuration
Save a optimization configuration
RUN
CANCEL

Choose Project: Saccharomyces

State Variables Set: Default set

Kinetic Parameters Set: Default set

Final Time: 50

Number of Iterations: 2000

Population Size: 200

Initialize Population: RANDOM

Interpolation Factor: 1000

dt: 0.005

State Variables

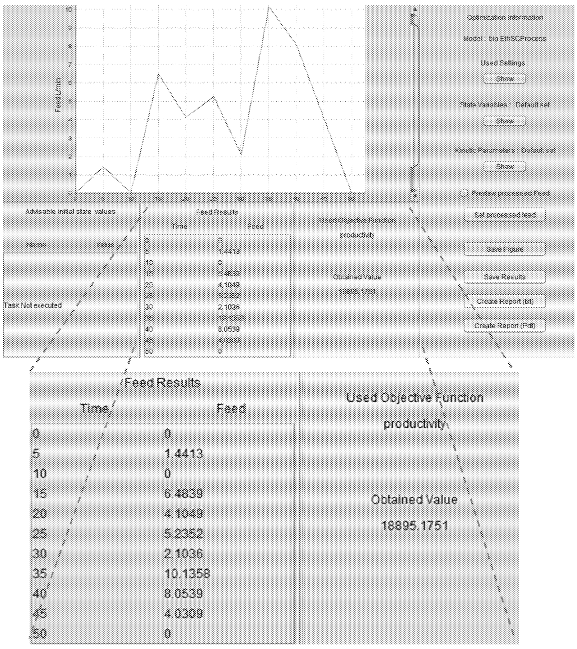
|  | Name | Value | Lower Limit | Upper Limit |
|---|---|---|---|---|
| X | 1.0 | 0 | 179.7698E308 |  |
| S | 150.0 | 0 | 179.7698E308 |  |
| P | 0.0 | 0 | 179.7698E308 |  |
| V | 10.0 | 0 | 200 |  |

Parameters

| Name | Value |
|---|---|
| YXs | 0.1 |
| SF | 150.0 |
| mu0 | 0.408 |
| Kp | 16.0 |
| Ks | 0.22 |
| pi0 | 1.0 |
| K'p | 71.5 |
| K's | 0.44 |

Figure 10: Graphical interface to execute optimization tasks.

After performing the optimization, the results are displayed as shown in Figure 11. A graph and a table are used to show the optimized feed trajectory. Information about the objective function is displayed, as well as the best value. The user can also see the parameters used in the optimization.



Feed Results

| Time | Feed |
|---|---|
| 0 | 0 |
| 5 | 1.4413 |
| 10 | 0 |
| 15 | 6.4839 |
| 20 | 4.1049 |
| 25 | 5.2352 |
| 30 | 2.1036 |
| 35 | 10.1358 |
| 40 | 8.0539 |
| 45 | 4.0309 |
| 50 | 0 |

Used Objective Function

productivity

Obtained Value

18895.1751

Figure 11: Results of the performed optimizations

## CONCLUSIONS AND FURTHER WORK

The aim of the *OptFerm* software was not to replace bioprocess optimization by trial-and-error approach, but to reduce the number of trials that are necessary to achieve the best results. So, with this tool the user is able to analyze the robustness of a fed-batch model, compare simulated data with experimental data, determine unknown parameters and optimize a feeding profile to be fed into a bioreactor.

The current software version has a major limitation: the absence of a graphical interface to visualize and edit models. This feature will be available in a future version. The user can still create the corresponding Java classes describing the model by differential equations and kinetic reactions. In future versions, functions for exporting/importing models in SBML (System Biology Markup Language) format will be implemented. Because OptFerm is implemented inside AIBench framework that has a plug-in concept, new functionalities or algorithms can be easily integrated.

## REFERENCES

Ascher,U.M. et al. (1997) Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Appl. Numer. Math.*, 25, 151–167.

Benjamin,K.K. et al. (2008) Genetic Algorithms Using for a Batch Fermentation Process Identification. *Journal of Applied Sciences*, 8, 2272–2278.

Chen,C. e Hwang,C. (1990) Optimal Control Computation for Differential-algebraic Process Systems with General Constraints. *Chemical Engin. Communications*, 97, 9.

Kawohl,M. et al. (2007) Model based estimation and optimal control of fed-batch fermentation processes for the production of antibiotics. *Chemical Engineering and Processing: Process Intensification*, 46, 1223–1241.

Mendes, R., Rocha, M., Rocha, I., Ferreira, E.C. (2006) A Comparison of Algorithms for the Optimization of Fermentation Processes. Proc. IEEE Conf. Evolutionary Computation, pp. 7371-7378, Vancouver, Canada, 2006.

Mendes,R. et al. (2008) Differential Evolution for the Offline and Online Optimization of Fed-Batch Fermentation Processes. In UK Chakraborty (ed.), *Advances in Differential Evolution*, ch.13, pp.299-318, Springer, 2008.

Pettinen,A. et al. (2005) Simulation tools for biochemical networks: evaluation of performance and usability. *Bioinformatics*, 21, 357–363.

Rocha, M., J. Neves, I. Rocha, E.C. Ferreira (2004) Evolutionary Algorithms for Optimal Control in Fed-batch Fermentation Processes, In LNCS 3005, pp.84-93, Coimbra, Portugal, Springer, 2004.

Rocha,M., Mendes, R., Maia, P. Pinto, J.P., Rocha, I., Ferreira, E.C. (2007) Evaluating Simulated Annealing Algorithms in the Optimization of Bacterial Strains. *Proc. EPIA 2007,* pp. 473-484, Springer, 2007.

Tzoneva,R. (2006) Method for Optimal Control Calculation of a Fed-batch Fermentation Process. *Proc. Mediterr. Conf. Control and Automation, 2006,* pp. 1–6.

Zhang,H. (2008) Optimal control of a fed-batch yeast fermentation process based on Least Square Support Vector Machine. *Intern. J. Engin. Systems Modelling Simulation*,1:63–68.

# MLPS: A METHOD FOR MODELING LIVESTOCK PRODUCTION SYSTEMS

Laurent Pérochon
Institut National de la Recherche Agronomique
UR1213 Herbivores
63122 Saint-Genès-Champanelle
France
E-mail: laurent.perochon@clermont.inra.fr

**KEYWORDS**
Beef Cow Cattle, Conceptual modeling, Unified Modeling Language, Livestock Production System.

## ABSTRACT

We present a method for modeling livestock production systems (MLPS), based on unified modeling language. A livestock production system is viewed as having four aspects: *production*, *decision*, *action* and *resource*. A three-stage process is used. First, the main interactions between the production system and external systems are modeled for each aspect by means of use case diagrams. Second, internal processes detail these use cases with refined use case, sequence or activity diagrams. Third, the model is built. Package and class diagrams model its structure, while sequence, activity, statechart and time diagrams represent its dynamics. We applied MPLS to a simplified beef production example.

## INTRODUCTION

The purpose of any livestock production system is animal production. Many models exist, focusing on the animal, the herd, or the whole farm system. Even with such scale differences, these models all simulate production (dairy or meat) through internal dynamics and structure. The use of a common method and formal representation would greatly simplify the construction of these models, and help to compare, develop and interconnect them.
Here we describe a method for modeling livestock production systems (MLPS). It uses the unified modeling language (UML) to formalize the model. First, we briefly present UML. MLPS is then detailed, along with the subset of UML used for the method. Finally, we apply MPLS to a simplified beef production example.

## UNIFIED MODELING LANGAGE

Unified modeling language is the international standard language in object-oriented software development (Rumbaugh et al. 2004). It aggregates (unified) main principles from more than 50 object-oriented methods of the 1990s. The Object Management Group, a consortium of major firms and institutes of the object community, normalized it in 1997. It affords 13 different types of diagram, each focusing on a particular aspect of the system. Although it was built for object software modeling, it can be used to model other types of domain (see for example Papajorgji and Pardalos 2006).

## DESCRIPTION OF MLPS

### Overview

The main goal of MLPS is to build a conceptual model of the studied system, represented by several UML diagrams. It is assumed that the objectives of the model have already been identified. MLPS can then be used in conformity with them.
The method models the whole system in terms of four aspects: *production*, *decision*, *action* and *resource*. *Production* covers every physical or flow process that builds or transforms products. *Decision* concerns either strategic or tactical management. It does not modify the system states. Humans or animals may take decisions. *Action* refers to the implementation of the decisions, and modifies the system's states. *Resource* represents what is utilized by the three other aspects and may be limited or unavailable at a particular time (for example forages).
MLPS refines each aspect model in a three-phase process, each phase focusing on a specific abstract level. This breakdown is important in the method because it prevents problems with several levels of abstraction being confounded. For example, it is not important to detail the growth process of the animal if the main functions of the whole farm system are not identified.

### The MLPS process

MLPS is composed of three phases by which the model is built up: *external*, *refine* and *build*.

It begins with the *external* phase, which models the exchanges between the production system and external systems. The issue here is which systems impact the production system, or are impacted by it. Using UML vocabulary, the external systems are actors, and the general description of the impact (exchange) is a use case. The exchanges are represented with UML use case diagrams (Figure 1).

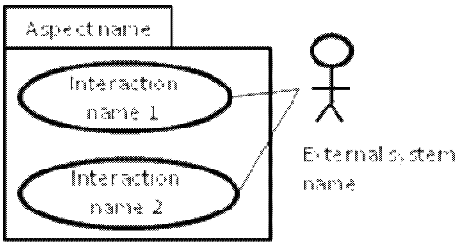


Figure 1: Use Case Diagram (stickman represents an actor)

At least one diagram for each aspect should be built. However, depending on model objectives, a particular aspect may have no external interactions.

At least one interaction must appear: the sale of the product or the final building of it (it then exits the system). This is the main interaction that drives the rest of the modeling process. If a use case must be further explained, a sequence diagram can be used (Figure 2).

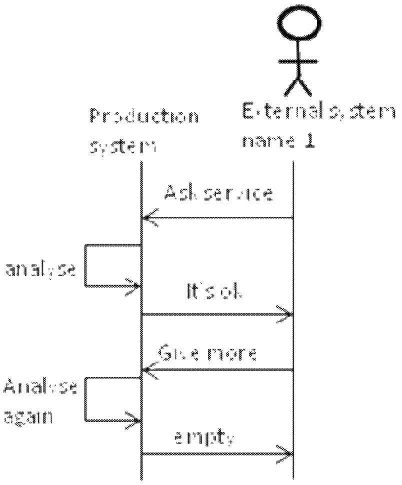


Figure 2: Sequence Diagram for a Use Case

The sequence diagram shows the sequence of exchanges between the production system and one or more external systems necessary to implement the use case. If the sequence is too long, the interaction can be modeled by means of scenarios, each represented by one diagram.

The next step is the *refine* phase. It concerns the internal process needed to implement the previously described external interaction. For livestock production systems, it may be, for example, breeding process, feeding management or dairy milking. This phase employs use case diagrams, linking previous use case with secondary use cases (internal process). During this stage, main secondary use cases may be detailed with an activity diagram (Figure 3).

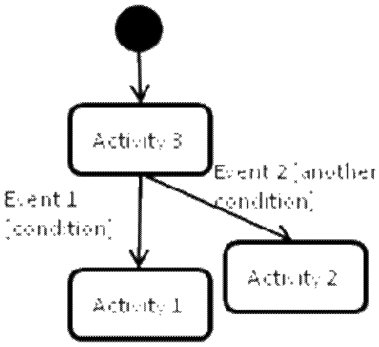


Figure 3: Activity Diagram

In Figure 3, Activity 1 will be implemented after Activity 3 if Event 1 happens and Condition is met. If time is important for a use case, timing diagrams may be used to show the system state changes over time. This type of diagram is not presented here owing to lack of space.

The last stage is the *build* phase. The first question here is: which internal parts of the system concern the use cases, and what are the links between these parts? For a livestock system they could be, for example, a farmer, a cow, and a culling action. Here the UML class diagram is used. It details the structure of the model. In UML, the internal parts are named classes (Figure 4).

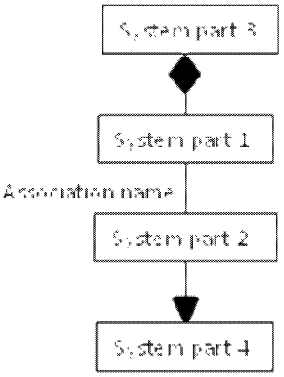


Figure 4: Class Diagram (▶ specialization link, ◆ composition link)

In Figure 4, System part 1 composes System part 3 and System part 2 is a sub-type of System part 4. System part 1 has a link (association) with System part 2. The package diagram can show whether the whole system is a set of subsystems in interaction. Each package represents a subsystem, and is made up of interacting classes (Figure 5).

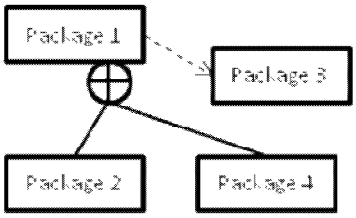


Figure 5: Package Diagram (⊕ aggregation, - - -> dependence link)

In Figure 5, Package 1 aggregates Packages 2 and 4, and depends on Package 3.

The next question concerns the necessary dynamics between the system's parts to implement a use case. Here the sequence diagram is best, but the activity diagram can also be used. To model state changes in internal parts, the statechart diagram is used. It is based on a representation similar to the activity diagram. For specific time transition modeling, timing diagrams are also possible. During this last stage, especially modeling the dynamics, the interaction between aspects appears.

## EXAMPLE OF A BEEF COW PRODUCTION MODEL

To illustrate the method, we present the use of MLPS for modeling an individual based simulator named SIMBAL (Pérochon et al. 2009). This simulator aims to predict the impact of farmer management on beef production. Two individual agents are modeled: the animal and the farmer. The diagrams presented have been simplified, retaining the important components.

### External Phase

In accordance with the model objectives and simplification choices, we kept only three aspects: *production*, *action* and *decision*. *Resource* is considered to be not limited or to have been included directly in production process equations. We first define external systems: client for the *decision* aspect and climate for the *production* aspect (Figure 6). No external interactions are modeled for the *action* aspect. We then choose main interactions to be modeled: *sell animal*, *breeding* and *growth*.

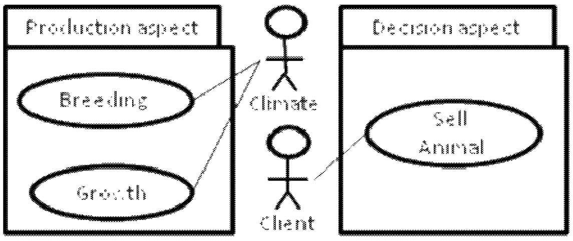


Figure 6: *External* Phase Use Case Diagrams

In our model, the climate interacts with the breeding and growth processes through level of feeding or seasonal effects on reproduction (Blanc and Agabriel 2008)

These diagrams are very simple but they are very important. At this high level of abstraction, choices have been made that will impact on the rest of the modeling process: some potential external systems and interactions are not modeled. What is represented is important but also what is not. For example, for the *decision* aspect, only one actor is present: the client who buys or asks for animals, while in reality, the bank, the family, agricultural government policies, etc. also interact.

### Refine Phase

Each external interaction is now detailed. As the most important is *sell animal* we present here the associated refined use case diagram (Figure 7). An internal process (secondary use case) appears: *animal production*. It concerns decisions taken to produce an animal. Among these, we consider that the main ones are *breeding* and *growth*.

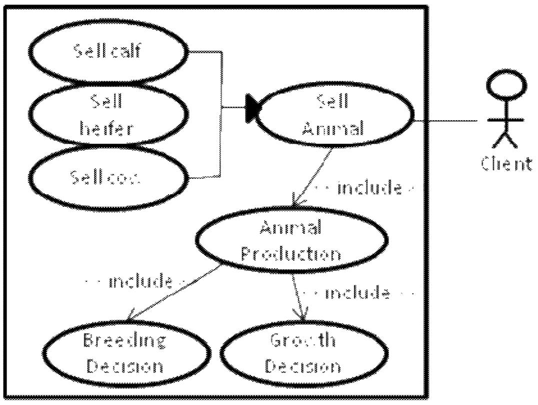


Figure 7: Refine Stage Use Case Diagram

Comparing Figures 6 and 7, we see the difference in levels of detail between the *external* and *refine* stages use case diagrams.

We considered that the *animal production* decision needed to be detailed for greater clarity. To do this we used an activity diagram (Figure 8).

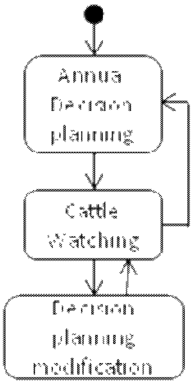


Figure 8: Activity Diagram for the Animal Production Use Case

This diagram introduces two new processes: watching the cattle and planning. These are the most important ones for the *decision* aspect. We can note that no actions on the cattle appear. This is normal because it is the goal of the *action* aspect. Figure 9 shows an activity diagram detailing the farmer's actions.

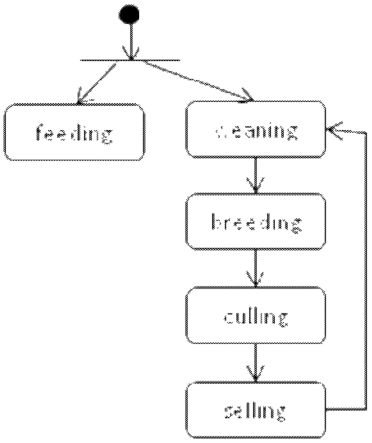


Figure 9: Activity Diagram for the Action Aspect

**Build Phase**

We now describe the internal structure of our model. First, instead of considering the model as a mere array of components, it is better to divide it into sub-systems; we use packages. The obvious way is to use aspects for our packages (Figure 10), three in our case study. But other sub-systems may exist.

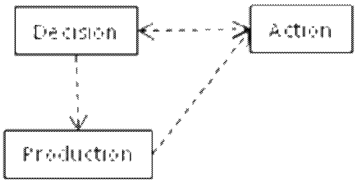


Figure 10: Package Diagram of the Aspects in the Model

In Figure 10, we see that *decision* depends on every other aspect, while *production* depends only on *action.* Each package is then detailed in one or more class diagrams. The central role of the farmer appears in the *decision* package (Figure 11). It takes decisions and orders them in a schedule. Several types of decision exist: culling, weaning, feeding, breeding and selling.

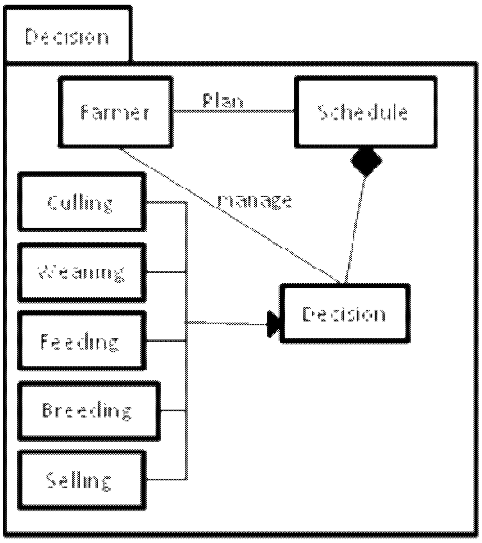


Figure 11: Class Diagram of the Decision Aspect

The main classes in the *production* aspect (Figure 12) are the animal and the batch (of animal).

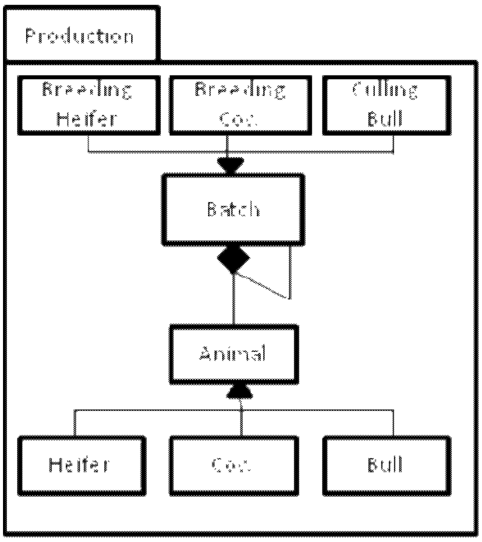


Figure 12: Example of Main Classes for the Production Aspect

To implement action or to take a decision, the farmer groups animals into batches (Ingrand *et al.*, 2002), each specialized for a specific task. For example, the *breeding heifer* batch concerns previously selected heifers kept for future breeding to renew the herd. The *culling bull* batch is males to be sold after fattening.

Now that the structure is modeled (though possibly not fully) the dynamics between model parts may be modeled. When a farmer renews animals in the *breeding cow* batch, he selects what he considers to be the best cows in conformity with his own criteria. This task uses parts of our model derived from each aspect (Figure 13).

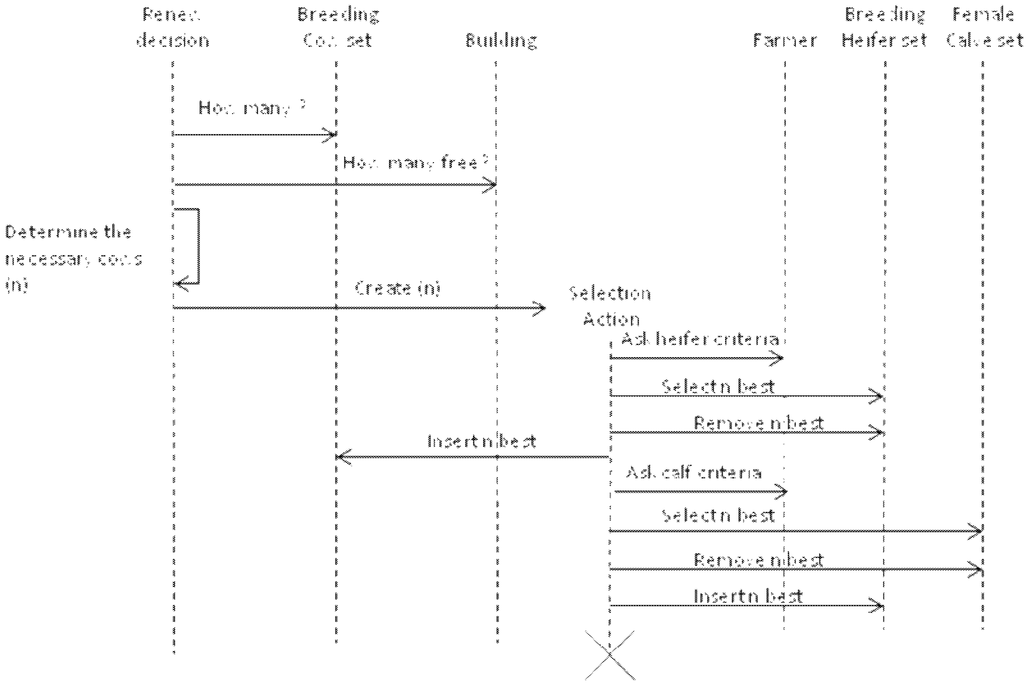


Figure 13: Example of Sequence Diagram Detailing Renewing Dynamic (Building is an example of resource)

*Renew decision* is in the *decision* aspect, *building* is in the *resource* aspect, *selection action* is in the *action* aspect and the others are in the *production* aspect. As breeding states are important in this system, we modeled it using a statechart diagram (Figure 14).

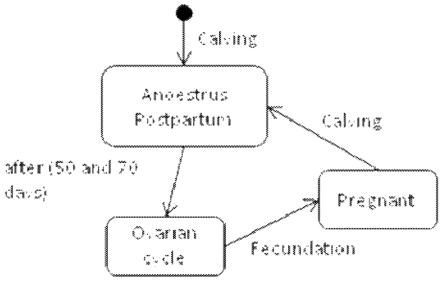


Figure 14: Statechart Diagram of the Cow breeding States

## CONCLUSION

The three important points of the method are (1) to model the system according to four specific aspects (*decision*, *production*, *action* and *resource*), (2) to split the analysis process into three stages (*external*, *refine* and *build*), each of which deals with a level of abstraction, and (3) to propose for each of them a UML subset. Using aspects to model an agronomical production system is not new. For example, Martin-Clouaire and Rellier (2003) used *manager*, *operative* and *productive* aspects. This method considers *resource* to be outside the operative system because managing resources is different from utilizing them. Van de Ven *et al.* (2003) chose *potential production*, *limited production* and *reduced production* aspects, which mainly covers our *production* aspect. Choosing a subset of UML was necessary because users often get lost among all the UML diagrams. This is why we give the subset together with suggestions about how to use it. MLPS approaches have been adapted for livestock production systems by the unified software development process method (Jacobson et al. 1999).

## REFERENCES

Blanc F. and J. Agabriel, 2008. "Modelling the reproductive efficiency in a beef cow herd: effect of calving date, bull exposure and body condition at calving on the calving-conception interval and calving distribution". Journal of Agricultural Science Cambridge, 146, 143-161.

Ingrand S.; B. Dedieu; J. Agabriel, L. Pérochon, 2002. "*Modélisation du fonctionnement d'un troupeau bovin allaitant selon la combinaison des règles de conduite. Premiers résultats de la construction du simulateur SIMBALL*". Renc. Rech. Rum. , 9, 61-64.

Jacobson I.; G. Booch; J. Rumbaugh. 1999. "the Unified Software Development Process". Addison Wesley.

Martin-Clouaire R. and J.P. Rellier. 2003. "Modélisation et Simulation de la Conduite d'un Système de Production Agricole". In *4e conf. de modélisation & simulation (MOSIM'03) (Toulouse, France)*. 699–704.

Papajorgji P.J. and P.M. Pardalos. 2006. "Software Engineering Techniques Applied to Agricultural Systems. An Object-Oriented and UML Approach." Springer. Applied Optimization.

Pérochon L.; S. Ingrand; C. Force; B. Dedieu; J. Agabriel. "SIMBAL: Simulation of a beef cattle herd". In *7e Workshop International: Modelling Nutrient Digestion and Utilization in Farm Animal.* (Paris, France).

Rumbaugh J.; I. Jacobson; G. Booch. 2004. "UML 2.0 Guide de Référence". Campus Press.

Van de Ven G.W.J.; N. de Ridder; H. van Keulen; M.K. van Ittersum. 2003. "Concepts in Production Ecology for Analysis and Design of Animal and Plant-Animal Production Systems". Agricultural Systems 76. 507-525.

# A PYTHON VALIDATION OF THE MULTILAYER DEVS THEORY: CASE OF A CATCHMENT BASIN

Emilie Broutin, Paul Bisgambiglia, Jean-François Santucci
University of Corsica
SPE UMR CNRS 6134
Quartier Grossetti, BP 52
F-20250 Corte
France
{broutin,bisgambi,santucci}@univ-corse.fr

**KEYWORDS**
DEVS, Simulation, Multilayer, Python,Natural systems

## ABSTRACT

Modelling and simulation of complex natural systems may involve incompatibilities when trying to perform data exchange between models defined by different domain specialists. During these exchanges some problems may appear according to the different types of data units or different time units used in the models involved in the overall modelling and simulation. This paper deals with a modelling scheme allowing to solve these problems. We will point out how we introduce different kinds of conversions functions into a special formalism based on DEVS called multilayer DEVS. We present also how we are validating this multilayer DEVS formalism through a Python implementation of the modelling and simulation of the hydrological behaviour of a catchment basin.

## INTRODUCTION

Modelling complex natural systems involves the cooperation of scientists from various horizons. Each of them is able to make his own model; each resulting model will represent a point of view of the studied natural system. Each model will involve the definition of its own data units. In order to fully study the behaviour of a complex natural system the previously defined models must have to communicate between them for exchanging data or anything else. However since these models are defined independently several problems may occur during the data exchange. These problems can invalidate the entire simulation.

We propose here a framework able to integrate the different models and to perform a safe simulation. This framework is based on the DEVS formalism created by professor Zeigler [1] [2] [3] [4] [5]. We called our formalism multilayered DEVS modelling and simulation. A detailed presentation can be found in [6]. DEVS is a formalism allowing to model a discrete event system. Two kinds of models are defined: 1) basic models from which larger ones are built, and 2) coupled models which describe how these models are connected together in hierarchical fashion. Basic models (called atomic models) are defined by the following structure:

CA = < X, S, Y, δint, δext, λ, ta> where:

(i) X is the set of input values;
(ii) S is the set of sequential states;
(iii) Y is the set of output values;
(iv) δint is the internal transition function dictating state transitions due to internal events ;
(v) δext is the external transition function dictating state transitions due to external input events ;
(vi) λ is the output function generating external events at the output, and
(vii) ta is the time-advance function which allows to associate a life time to a given state.

The behaviour of an atomic model is illustrated as follows: the external transition function describes how the system changes state in response to an input. When an input is applied to the system, it is said that an external event has occurred. The next state s' is then calculated according to the current state s. The internal transition function describes the autonomous (or internal) behaviour of the system. When the system changes state autonomously, an internal event is said to have occurred. The next state s' is therefore calculated only according to the current state s. The output function generates the outputs of the system when an internal transition occurs. The time advance function determines the amount of time that must elapse before the next internal event occurs, assuming that no input arrives in the interim.

An atomic model enables us to specify the behaviour of a basic element of a given system.

A coupled model indicates how to couple (connect) several component models together to form a new model. This latter model can itself be employed as a component of a larger coupled model, thus giving rise to hierarchical construction. A simulator is associated with the DEVS formalism in order to execute a coupled model's instructions so as to actually generate its behaviour. The architecture of a DEVS simulation system is derived from the abstract simulator concepts (Zeigler and al. 2000) associated with the hierarchical and modular DEVS formalism. The abstract simulator allows the definition of a simulation tree whose root element is dedicated to the time advance management.

The rest of the paper is organised as follows. In the second section we present the multilayered DEVS formalism. Section 3 deals with the example used in order to validate the proposed formalism. We will describe two models involved in the modelling of the behaviour of a catchment basin which have been defined independently by different modellers. The implementation and validation of the multilayered DEVS formalism will be described in detail in section 4. In the final paper we will present the obtained results which are going to be generated in the next weeks. Finally section 5 will briefly summarized some conclusions and will present future work.

## THE MULTILAYERED DEVS FORMALISM

The multilayered DEVS formalism allows the modelling of a complex natural system by the integration of several kinds of models. These models created by different domain specialists are called behavioural models. In order to perform the integration of these models we have defined a DEVS model called Assembly Model. This coupled model is the central element of the proposed formalism. The Assembly model will treat every data shared by the models. This special component is a coupled model composed of two following main kinds of atomic models:

(i) The Driven models; these models are in charge of the data transmission. Each data shared between the behavioural model pass through the Assembly model. After received the data, a write order is send to the corresponding Storage models (describe below); furthermore the second role of the driven models is to transmit to an associated behavioural model the data received from the storage models. Each behavioural model is link to one and only one Driven model.

(ii) The Storage models; their role is to register the data shared between models. There is a storage model for each type of data.

The Assembly model also contains conversion functions. There are two kinds of functions:

(i) Details conversion functions: these functions act when there are some details scale problems (units problems); for example if a model use km as unit and another one cm, a function converts the data to the right units. These functions insure the validity of the data.

(ii) Temporal conversion functions: theses functions are based on the Jerome Euzenat's theory [9]. These functions allow us to not redefine the classical DEVS simulator.

## THE VALIDATION EXAMPLE: A CARTCHMENT BASSIN

A catchment basin can be decomposed into 2 different models: the hydrological model and the snow model

### Hydrological model

GR3J is an hydrological model for the study of catchment. It performs good results by using a representation of the rainfall-runoff process as simple as possible and depending on very few parameters.

A complete description of GR3J can be found in [11].

Figure 1 represents the structure of the GR3J model. P and E are respectively the precipitations and the potential evaporation; the first transformation takes place in a reservoir called "interception" its capacity is null.

There are two functions:

- Production function: the soil reservoir which is defined by its capacity noted down here A and its

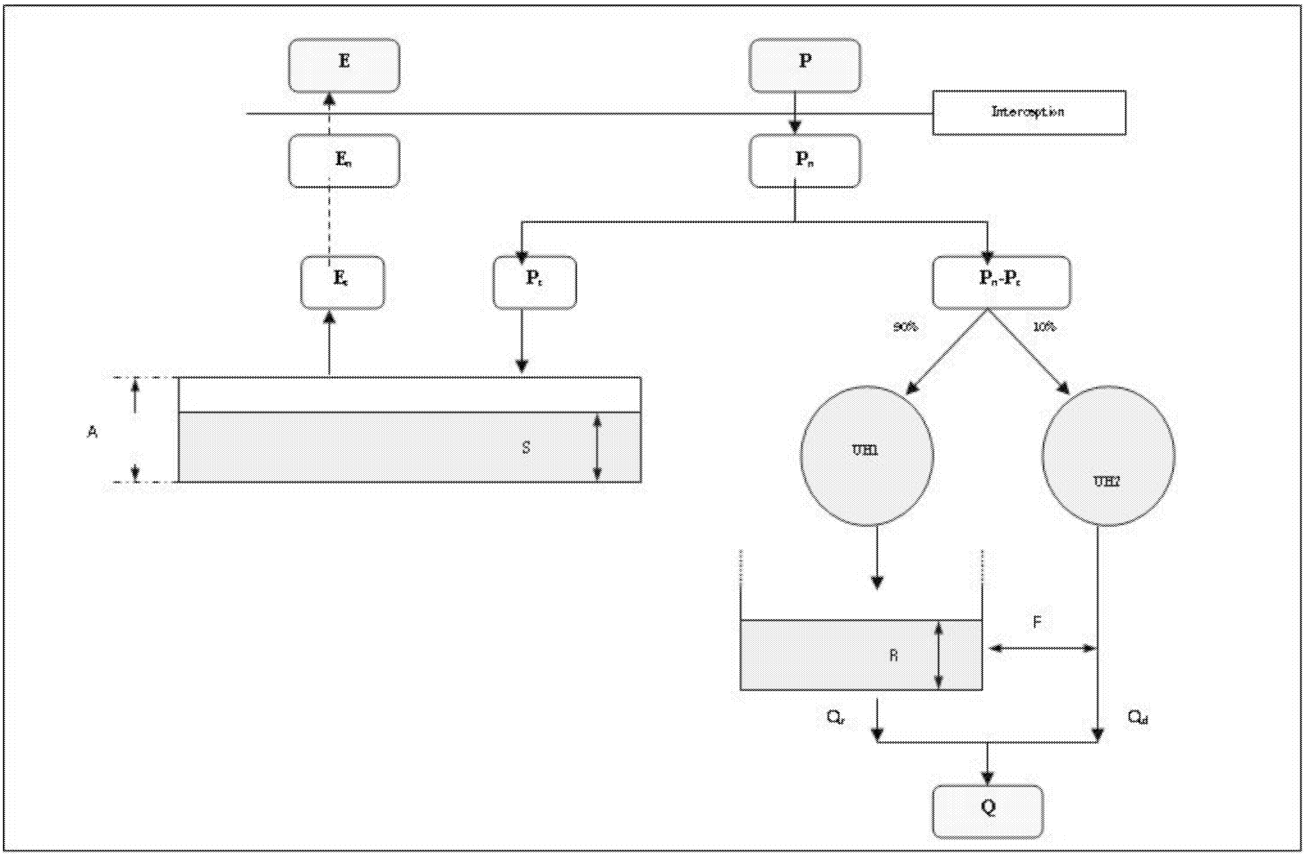


**Figure 1 GR3J model**

real level S. S evolves with the rain Pn and the evaporation En. The input (Ps) and output (Es) flow take place when Pn and En are positive.

- Transfer function: the water which does not go to the soil reservoir represents the available water for runoff (Pn-Ps). This water is divided into two parts: the most important is the left part (90% is transformed by a 1-day unit hydrograph (UH1) while 10% is transformed by a second unit-hydrograph (UH2)). The first part, after routing by UH1 is given as input to a reservoir whose storage is R. This reservoir is subject to an exchange of water F. Q represents the daily stream flow.

**Snow model**

For this model, the air temperature will be very important. In fact we can determine if there is some rain or some snow with the air temperature. Let us call the air temperature TA, the critical temperature below which it is snowing are call TLow. If TA is higher than temperature TUp it is raining. If temperature is between TLow and TUp it gives a mix of rain and snow. The reader will find below the corresponding equations.

$$f = 1 \quad \text{if } T_A \geq T_{UP}$$

$$f = (T_A - T_{LOW}) / (T_{UP} - T_{LOW}) \quad \text{if } T_{LOW} \leq T_A \leq T_{UP}$$

$$f = 0 \quad \text{if } T_A \leq T_{LOW}$$

Snowmelt is calculated with these equations:

$$M = K_M * (T_A - T_{B,M})$$

$$K_M = K_{MIN} * (1 + K_{CUM} * M_{CUM})$$

Where M is the snowmelt, K the degree-day, $T_A$ the average air temperature, $T_{B,M}$ is temperature in which the snow melting. $M_{CUM}$ is the melting accumulated during the season, $K_{MIN}$ is the minimum of the degree-day, $K_{CUM}$ is a parameter for some calibrations.
More details of this model can be found in [12].

## THE IMPLEMENTATION

We have converted these models to DEVS model in order to use them with our multilayer framework.

Hydrological model in DEVS

The hydrological model can be converted into DEVS formalism. Figure 2 represents the coupled DEVS model derived from GR3J. All of the submodels are atomic models:

- Dispatcher handles the water distribution among the two unit-hydrographs
- Delay1 and Delay2 represent the unit-hydrographs. They calculate the amount of water to deliver and the associated delay.
- Cumul generates the streamflow, depending on the water delivered by Delay1 and Delay2.

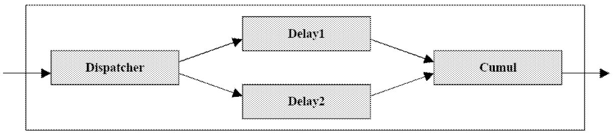


**Figure 2 GR3J hydrological model in DEVS**

**Snow model in DEVS**

This model can easily be transform in an atomic model.
The basic principle is the following one:
When an input is received ($\delta_{ext}$ function), the variables are updated, and the current state of the model is computed. The model is active in the following cases:

- The available liquid water is higher than the quantum
- The sum of liquid water and snow amounts is higher than the quantum and the current temperature allows snowmelt.

Then, the time advance function (ta) calculates the time before the next job to execute:

- If the model is in the passive state, this time is infinite
- If the available amount of water is higher than the quantum, the output is quantized.
- If snowmelt is necessary to have a water amount equal to the quantum, the time required to have a sufficient snowmelt is calculated

The output function ($\lambda$) generates an output equal to the quantum. The internal transition function ($\delta_{int}$) updates the model variables. If snowmelt is necessary to deliver a quantum amount of water, an output is planned for later, but we have taken into account the possibility for an input event to occur in the meantime. If necessary, the external transition function calculates the snowmelt for the given period.

These two models represent the views of two specialists; they are the behavioral models. They both link to the Assembly model.

We have implemented both the multilayered formalism and the DEVS coupled models and atomic models involved in the validation example of section 3 using the PythonDEVS simulator [7,8]. The Python-DEVS Modelling and Simulation package provides an implementation of the standard classic DEVS formalism described in section 1. The package consists of two files, DEVS.py and simulator.py. The first one provides class architecture that allows hierarchical classic DEVS models to be easily defined by subclassing the AtomicDEVS and CoupledDEVS classes. The Simulator engine (SE) is implemented in the second file. Based on the principles of simulation describe in section 1, it allows to perform discrete event simulation. Even if the PythonDEVS software involves a simulation engine which offers limited means to terminate a simulation and provides no easy model-reinitialisation possibilities we have been able to use it in order to efficiently test our approach.

## RESULTS

The two models presented below have been transformed onto DEVS models and been linked through the assembly model, this section presents two curves representing some results. The curves (Figure 3) represent the comparison between the real observed measures and the computed measures. The first one presents results during a year and the second the results for 8 years.

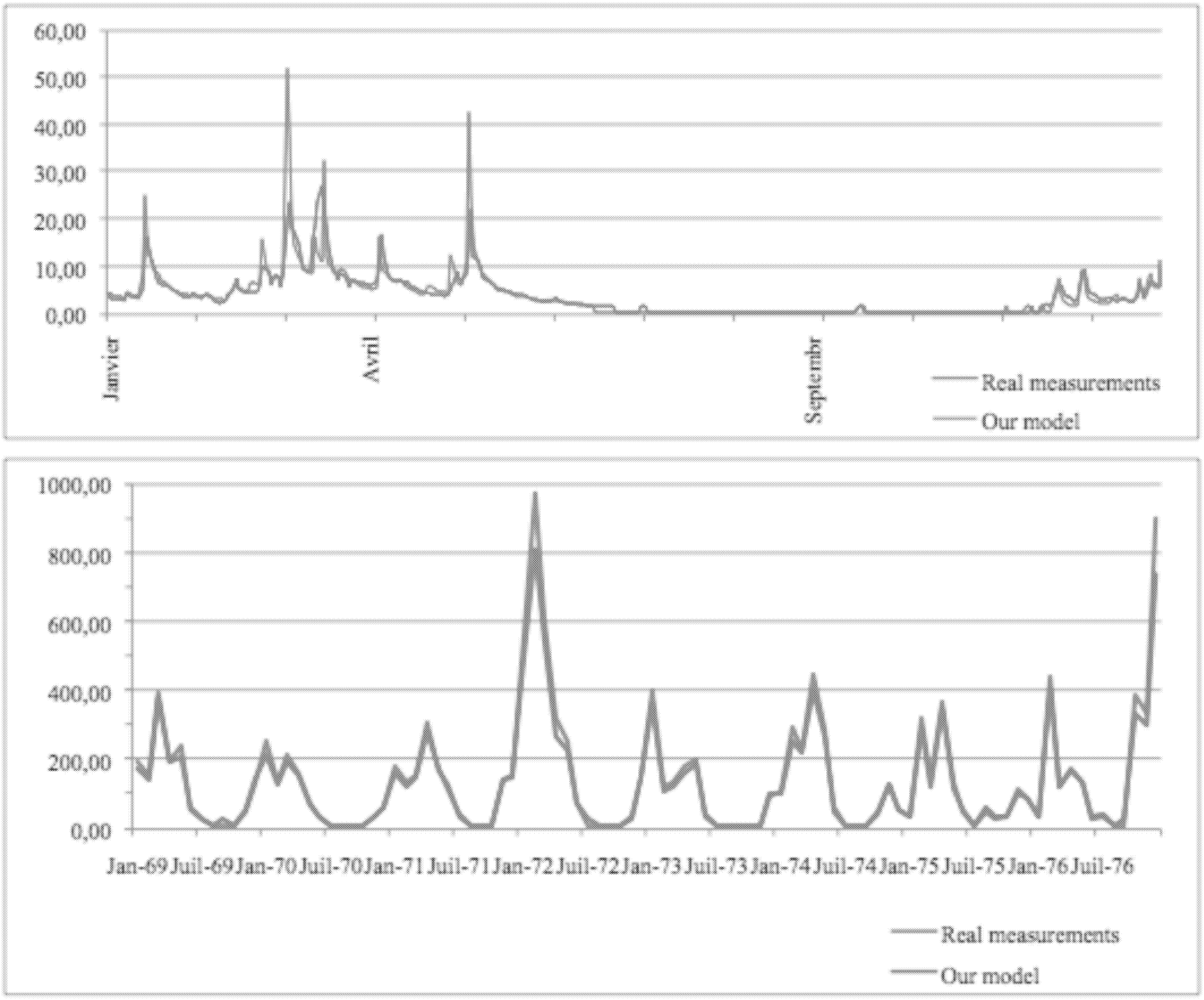


Figure 3 Results :comparison between the classical approach and our approach

We can see some differences between the two curves (red and blue), the first one (the blue one) show the results takes by a specialist. The red curve, presents the results obtain with our catchment model. We can see that there's a difference between our results and the "real" results; we can explain that: our model is build from the GR3J model, some parameters must be defined as a constant because of its complex nature, so some details are not taking into account. Moreover as its explain in the GR3J presentation, the models is not provide for deal with flood. During a flood a lot of water runoff in a very short time so GR3J model do not taking into account that, the flood is a special event that must be treated apart. Furthermore we have pointed some inconsistencies in the measured results that can explain some little differences between our results.

Despite we can see that out results is very near from the real measures, we can predict quite faithfully the behaviour of a catchment basin. The second set of curves show this fact even better. Here we cover 8 years from 1969 to 1976, we can see here that our results even closer to reality.

## CONCLUSION

We have presented here a way to modeling complex natural system. We use the multilayered DEVS formalism [6] [10]. We have implemented both the Assembly coupled model and the DEVS models involved by the validation example dealing with the behavior of a catchment basin. The validation example allows us to fully illustrate and test the exchange and sharing of data between models that do not have the same data unit and the same time unit. We also made two curves to show the reliability of our results. These curves point out that our results are really close from the real measurements especially when the scale covers a large period.

Furthermore we are also implementing a complex application dealing with forest fire simulation using the multilayered DEVS formalism. We plan to develop a complete comparison between the simulation of forest fires using a classical DEVS model and simulation of forest fires using the multilayered DEVS formalism.

## REFERENCES

[1] Zeigler, B.P. 1975. "Theory of Modelling and Simulation" *Academic Press.*

[2] Zeigler, B.P. 1976. "Theory of Modeling and Simulation." *New York, Wiley.*

[3] Zeigler, ,B.P. 1984. "Multifaceted Modelling and Discrete Event Simulation". *London, Academic Press.*

[4] Zeigler, B.P. 1990. "Object-Oriented Simulation with Hierarchical, Modular Models".

[5] Zeigler, B.P. , H.S. Sarjoughian. 2000. "Creating Distributed Simulation Using DEVS M&S Environment".

[6] Broutin, E. P, Bisgambiglia. JF, Santucci. "Simulation of heterogeneous DEVS models ; application to the study of natural systems". In proceeding of Spring Simulation Conference San Diego CA.

[7] Bolduc J.S. and Vangheluwe H, PythonDEVS: a modeling and simulation package for classical hierarchical DEVS. Technical Report, MSDL, McGill University, 2001

[8] http://moncs.cs.mcgill.ca/MSDL/reasearch/DEVS

[9] Euzenat, J. 1994. Granularité dans les représentations spatio-temporelles. Rapport de recherches

[10] Broutin E, Bisgambiglia P, Santucci JF*, Multilayered and heterogeneous modeling and natural complex systems, European Simulation and Modelling Conference, OCT 27-29, 2008 European Technol Inst, Havre, FRANCE, EUROPEAN SIMULATION AND MODELLING CONFERENCE 2008, Pages: 338-342 Published: 2008*

[11] Edijatno, Nilo de Oliveira Nascimento, Xiaoliu Yang, Zoubir Makhlouf & Claude Michel. 1999. *GR3J: a daily watershed model with three free parameters*. Hydrological Sciences Journal, 44(2), pp. 263-277.

[12] Vehviläinen, B. 1992. *Snow cover models in operational watershed forecasting*. Publications of Water and Environment Research Institute 11, PhD thesis. Helsinki. 112 p.

## BIOGRAPHIES

**Emilie Broutin** is a PhD student in computer science in the University of Corsica. Her current research focuses on DEVS multilayered modeling and simulation.

**Paul Bisgambiglia** is a professor in Computer Sciences at the University of Corsica. He is responsible of the modeling and simulation team of the UMR CNRS 6134. His research activities concern the techniques of modeling and simulation of complex systems and the test of systems described at high level of abstraction. He makes his researches in the laboratory of the UMR CNRS 6134.

**Jean-Francois Santucci** is Full Professor in Computer Sciences at the University of Corsica since 1996. His main research interest is Modelling and Simulation of complex systems. He has been author or co-author of more than 100 papers published in international journals or conference proceedings. He has been the scientific manager of several research projects corresponding to European or industrial contracts. Furthermore he has been the advisor or co-advisor of more than 20 PhD students and since 1998 he has been involved in the organization of more than 10 international conferences. He is conducting newly interdisciplinary researches involving computer sciences, archaeology and anthropology: in the one hand he is performing researches in the archeaoastronomy field (investigating various aspects of cultural astronomy throughout Corsica and Algeria using tools issued from Computer Sciences) and on the other hand he is applying computer sciences approaches such as GIS (Geographic Information Systems) or DEVS (Discrete EVent System specification) to anthropology.

# MODULAR SIMULATION AND DESIGN

# A SOFTWARE COMPONENT WHICH GENERATES REGULAR NUMBERS FROM REFINED DESCRIPTIVE SAMPLING

Megdouda OURBIH – TARI, Abdelouhab ALOUI and Amine ALIOUI
Laboratory of Applied Mathematics
University A. Mira of Bejaia
Algeria
E-mail: { megtari | aaloui_abdel } @yahoo.fr

**KEYWORDS**
Random sampling, Descriptive sampling, Refined descriptive sampling, Software component, Simulation.

## ABSTRACT

In this paper, we propose a software component that implements a generator of regular numbers from primes when required by the simulation using refined descriptive sampling. The latter is regarded as the best sampling method. In order to validate the proposed component, an illustration of the uniformity is given together with the simulation of an M/M/1 queuing system. The simulation results were compared to those obtained using the generator rand () included in the C programming language under Linux. The best results are given by the proposed software component.

## INTRODUCTION

Nowadays, simulation covers significant challenges in all areas of engineering (technical, commercial, financial ...) because it is a necessary aid to decision making and control of accuracy. When all else fails, then simulate. Monte Carlo simulation is a sampling experiment based on the succession of a large number of random draws. This method is well known and intensively used; nevertheless, it is still a research subject in three main areas:

1. The quality of the random generators used (Makato and Takukuji 1998).
2. The techniques of variance reduction (Henry and Flora 1998),
3. The techniques of behavioural model building (queuing network, stochastic Petri network...) that seek help to better formalize some problems before being processed.

But, its success was harmful. Indeed, Monte Carlo method was provided to solve any kind of problem. No study has been made for the type of problem for which it was particularly adapted. Nowadays, it is also used to generate the initial solution of other simulation techniques (taboo search, simulated annealing...). The simulation results obtained through a Monte Carlo method are of modest accuracy, this is due to the extreme slow convergence. Because of its limits, a new paradigm emerged: it is not always necessary to resort to randomness. Then, non-random sampling methods were derived from this paradigm. Descriptive sampling (DS) (Saliby 1990) and refined descriptive sampling (RDS) (Tari and Dahmani 2005a) are both with this paradigm. The efficiency of RDS over DS and RS is proved by several comparisons, on a flow shop system and a production system (Tari and Dahmani 2005b, 2005c).

This paper is in the quality of number generators used in a simulation. A software component is developed implementing the RDS method for the generation of input samples for simulators. The choice of the method is motivated by the supremacy of its quality (Tari and Dahmani 2006). An M/M/1 Simulator is also developed in order to validate the proposed software component and the performance measures of a simple queuing system are established for comparison with the C-language random number generator under Linux.

## ALTERNATIVES TO MONTE CARLO SIMULATION

An algorithmic random number generator must satisfy a set of criteria such as:

Uniformity: The number stream shall pass the tests of a uniform distribution.

Independence: The full orbit and the particular sub-orbits must be independent.

The extended period: Given that the typical programs have execution times running from several hours to several months, the simulation programs running on supercomputers need random numbers. This imposes a lower limit to the period of generators.

Reproducibility: checking programs during their development, we must be able to reproduce exactly the generated stream of random variables

Portability: For reasons of verification, it is sometimes necessary to run the program on different machines, possibly with different word lengths.

Separate sub-streams: If a simulation is performed on a multiprocessor machine or if the computation is distributed across a network, then, the sub-stream used in each sub-task of the program must be independent.

Effectiveness: Since the call to the routine of the generator is done many times, it is then necessary that its program is as simple as possible with required minimal operations.

### Refined descriptive sampling

To reduce the risk of bias, RDS procedure was proposed as an alternative approach to Monte Carlo Simulation. This method is mainly concerned with a block of prime numbers which must be situated inside a generator aiming to distribute regular samples of prime size when required by the simulation. Compared to DS this approach removes the need to determine in advance the sample size.

Let $p_q$, q=1,2,3,... be distinct prime numbers and a simulation experiment of M replicated runs, terminates when $m_M$ prime numbers have been used, which derives $m_M$ sub-runs. In this procedure, we present regular samples from $p_q$ then $p_{q+1}$ and so on for any q in random order as required by the simulation. We terminate when the simulation terminates. Using RDS, subset values for the input random variable X are generated as required by the simulation.

The general method of the inverse transform produces regular subset values given by

$x_i=F^{-1}(r_i)$ for $i=1,2,...,p_{qj}$, $q=1,2,...,m_j$ and $j=1,2,...,M$
where
$F^{-1}(r)$, $r \in [0,1]$ is the inverse cumulative input distribution and $r_i=(i-0.5)/p_{qj}$ $i=1,2,...,p_{qj}$, $q=1,2,...,m_j$ and $j=1,2,...,M$

and the sequence of each regular subset $\{r_i, i=1,2,...,p_{qj}\}$ is randomised.

## PRESENTATION OF THE DEVELOPPED COMPONENT

A generator aiming to distribute regular samples of prime size, using RDS, when required by the simulation is developed. We illustrate the developed component in two figures representing the uniformity distribution of generated points. A set of 5000 regular numbers are generated by the developed component RDS, and represented over a plane (fig 1). Furthermore, the set of numbers is divided into two sub-sets of 2500 points which are represented over another plane (fig 2).

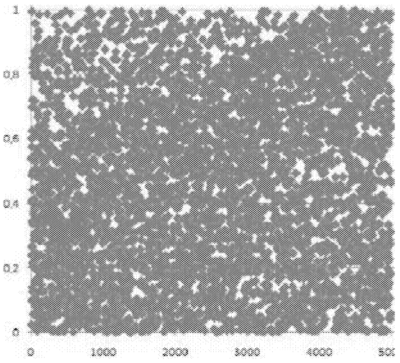


Figure 1: Graph of 5000 numbers obtained by the use of get RDS showing the uniformity

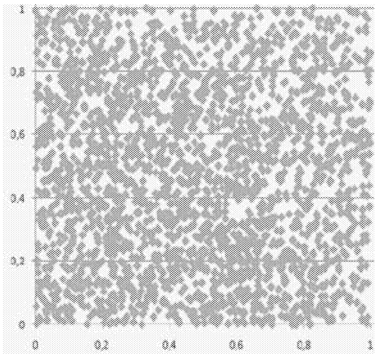


Figure 2: Graph of pairs of 2500 numbers obtained by the use of getRDS showing the uniformity

### Description and implementation

The main file of the developed library, called "getRDS" contains the following four main functions:

alea_min_max (): This function generates a random integer between the respective integer MIN and MAX such as MIN <MAX <4294967295, these integers will be introduced by the user. MIN and MAX represent respectively the minimum and maximum.

Is_Prime() : The aim of this function is to check if an integer N is a prime or not.

pgetrand() : This function generates a prime number randomly between the respective MIN and MAX such as MIN <MAX <4294967295, these integers will be introduced by the user.

ugetRDS() : The objective of this function is the generation of uniform numbers using RDS method.

The design of the developed component is special in two ways. First, the programmer can afford the service of each function individually or separately and is useful as all parameters are fixed by the user.

There are two possibilities for using this library, do either:

# include "RDS.h": If this file is copied into the same directory of the program that it uses.
or
# include <RDS.h>: If the file is copied into the directory standard library of C language under linux / usr / include.

In the first case, we will have a program of the following structure

```
#include <stdio.h>
#include "RDS.h"
int main() { }
```

But, in the second case, we will have a program of the following structure

```
#include<stdio.h>
#include <RDS.h>
```

```
int main() { }
```

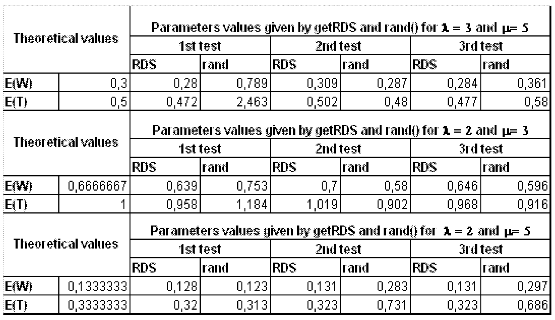
### The use of the developed component

Let's show how to use the developed component. We create a program under the name example.c and the following source code:

```
# include <stdio.h>
# include<RDS.h> / * call the component, here the component is assumed to be recorded under the directory / usr / include / * /
int main ()
{
int i;
printf ( "A stream of 5 integer random numbers between 2 and 50:");
for (i = 0; i <5; i + +)
printf ( "% d", alea _min _max (2,50));
printf ( "A stream of 5 prime numbers between 3 and 50:");
for (i = 0; i <5; i + +)
printf ( "% d", pgetrand (3,50));
printf ( "A stream of 5 regular numbers between 0 and 1:");
for (i = 0; i <5; i + +) \ \
printf ( "% 0.3f", ugetRDS (3,50));
if (Is_premier (11)) printf ( "11 is a prime number.");
else printf ( "11 is not a prime number.");
return 0;}
```

### An example of application

The developed component was validated by simulating a simple queuing system of type M/M/1. The same queuing system was also simulated using the random number generator rand () included in the C language under Linux. The considered input parameters of the studied queuing system are: The rate service time, μ, and the arrival rate, The output parameters are the mean waiting time (E(W)) and the mean stay time (E(T)). Then, an M/M/1 Simulator is developed and implemented with C-language under Linux, in order to validate the "getRDS" software component and to compare it with the random number generator rand() integrated by default in the C-language. The design of the simulator is special in two ways. First, it affords the comparison of both sampling methods and is useful as all parameters are fixed by the user. Given these input parameters, the simulator outperforms M=100 iterations for each sampling method to compute both performance measures of the system being studied. The results are given in the following table 1.



| Theoretical values | | Parameters values given by getRDS and rand() for λ = 3 and μ= 5 | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1st test | | 2nd test | | 3rd test | |
| | | RDS | rand | RDS | rand | RDS | rand |
| E(W) | 0,3 | 0,28 | 0,789 | 0,309 | 0,287 | 0,284 | 0,361 |
| E(T) | 0,5 | 0,472 | 2,463 | 0,502 | 0,48 | 0,477 | 0,58 |
| Theoretical values | | Parameters values given by getRDS and rand() for λ = 2 and μ= 3 | | | | | |
| | | 1st test | | 2nd test | | 3rd test | |
| | | RDS | rand | RDS | rand | RDS | rand |
| E(W) | 0,6666667 | 0,639 | 0,753 | 0,7 | 0,58 | 0,646 | 0,596 |
| E(T) | 1 | 0,958 | 1,184 | 1,019 | 0,902 | 0,968 | 0,916 |
| Theoretical values | | Parameters values given by getRDS and rand() for λ = 2 and μ= 5 | | | | | |
| | | 1st test | | 2nd test | | 3rd test | |
| | | RDS | rand | RDS | rand | RDS | rand |
| E(W) | 0,1333333 | 0,128 | 0,123 | 0,131 | 0,283 | 0,131 | 0,297 |
| E(T) | 0,3333333 | 0,32 | 0,313 | 0,323 | 0,731 | 0,323 | 0,686 |

Table 1: Summary of an M/M/1 simulation results from different tests of 100 replicated runs using "getRDS" and rand() generators

## CONCLUSION

We have proposed and implemented an RDS generator, then we have shown how it works and an example of application was given through an M/M/1 simulator. As shown in table 1, the experimental results demonstrate that all performance measures of the M/M/1 queuing system using the "getRDS" component are closer to the theoretical values than those obtained by rand() function. Therefore, these results strongly support the efficiency of RDS. The "M/M/1 simulator" affords a multitude of simulation experiments and shows that even with more than one output variable observed through simulation, the results from RDS are better than RS. Given, that rand () is a very good random number generator, this confirms that our getRDS component really is a good generator.

## REFERENCES

Henry .S, Flora K., 1998. "Variance Reduction Techniques for Large Scale Risk Management", *In H. Niederreiter and J.Spanier: Monte Carlo and Quasi Monte Carlo Methods,* pages 419-433.

Makato. M, Takukuji. N., 1998. "Dynamic Creation of pseudorandom Number Generators", *In H.Niederreiter and J.Spanier: Monte Carlo and Quasi Monte Carlo Methods,* pages 56-69.

Saliby, E. 1990. "Descriptive Sampling: A better approach to Monte Carlo simulation". *Journal of the Operational Research Society*, 41, 12, 1133-1142.

Tari, M. and Dahmani, A. 2005. "The refining of descriptive sampling." *International Journal of Applied Mathematics & Statistics*, 3, M05(march), 41-68.

Tari, M. and Dahmani, A. 2005. "Flowshop simulator using different sampling methods." *Operational Research: An International Journal,* 5, 2, 261-272.

Tari, M. and Dahmani, A. 2005. "The three phase discrete event simulation using some sampling methods" *International Journal of Applied Mathematics & Statistics*, 3, D05 (Dec) 37-48.

Tari, M. and Dahmani, A. 2006. "Refined descriptive sampling : A better approach to Monte Carlo simulation." *Simulation Modelling Practice and Theory*, 14, 143-160.

# Composition of product-form Generalized Stochastic Petri Nets: a modular approach

Simonetta Balsamo and Andrea Marin
Dipartimento di Informatica
Università Ca' Foscari di Venezia
Via Torino, 155
Venice, Italy
{balsamo,marin}@dsi.unive.it

**KEYWORDS**
Stochastic modeling, product-form, exact analysis

## ABSTRACT

In this paper we present a novel approach to specify and analyze complex system using product-form models. The main strengths of this approach are its high modularity and its ability of dealing with a very large class of product-form models. This has been possible because the product-form analysis is based on two properties that are formulated at a very low level, i.e., the Markov implies Markov property and the Reversed Compound Agent Theorem. We propose a unifying framework for combining product-form models defined in terms of different formalisms and we give the conditions that allow the composition to be in product-form. The semantic of their combination is formally defined because the various sub-models are transformed into GSPNs with an equivalent underlying process. In particular, we illustrate with several examples that we can perform analysis of models with non-linear traffic equations, including those with some components being G-queues, product-form stochastic Petri nets, or multi-class queueing stations.

## INTRODUCTION

Stochastic models have proved to play a pivotal role in the performance analysis of software and hardware architectures. The model of the system can be defined according to a large set of formalisms which ranges from Petri nets extensions to queueing systems and others. Generalized Stochastic Petri Nets (GSPNs) [Marsan et al. (1995)] are a well-known formalism capable of representing complex systems in a formal way. This formalism is a stochastic extension of Petri Nets (PNs) that had been introduced to describe systems with parallel computations. Informally, PNs consist of places, transitions and arcs that connect places to transitions or viceversa. Tokens represent the state of the model and are associated with the places. The firing of the transitions change the state of the model. In the Stochastic Petri Nets (SPNs) the transition firing takes an exponentially distributed random time and, under a set of assumptions on the firing semantic, the marking process, i.e., the stochastic process that describes the state of the model along the time, is a Continuous Time Markov Chain (CTMC). GSPNs [Marsan et al. (1995)] can be seen as a extension of SPNs that admits two types of transitions: immediate and timed. The firing of the former ones occurs in a deterministically zero time, while the firing of the latter ones requires an exponentially distributed random delay. We summarize the main strengths of GSPNs.
- It has a strong semantic. Indeed, given a GSPN model with its initial marking, the underlying stochastic process is uniquely determined. This property is not shared with all the formalisms for the stochastic modeling, e.g., queueing networks are usually described by a high level language.
- It allows for qualitative analysis of the system, by the so-called structural analysis, e.g., using the net invariants.
- The state of the model and its structure are strongly separated. For instance, we can define a structurally finite model with an underlying process with infinite states.
- The formalism, with inhibitor arcs, is very expressive. Indeed, it is has been proved that PNs with inhibitor arcs are Turing-complete.

If the process underlying a GSPN has a steady state, then we can compute its stationary probability distribution. This plays a pivotal role in the performance evaluation field, because from the stationary distribution of a model we can derive a set of significant performance indices, such as the throughput, the response time distribution, the distribution of the number of tokens in a place, and so on. However, the analysis of a model defined in terms of GSPNs may easily become an unfeasible task. This is mainly due to two reasons: the first problem is shared with all PNs models, i.e., it is computationally expensive (when not impossible) to build the set of all the reachable states of the model given its initial state. Indeed, it is known that the reachability problem (given an initial marking, is a marking reachable after any number or sequence of transition firing?) for PNs without inhibitor arcs is EXPSPACE, while it is equivalent to the halting problem of the Turing machines for PNs with inhibitor arcs. The second problem concerns the calculation of the performance indices when the model admits a steady-state (i.e., when the underlying CTMC is ergodic). Indeed, even small models may have huge state spaces and, in the general case, the stationary state probability distribution is calculated as the solution of a linear system whose rank is the number of states of the model. Usually, this is computationally expensive and may soon lead to numerical instability of the algorithms. These problems are partially overcome by product-form models. These admit a decomposition in a set of interacting sub-models whose stationary distribution can be computed in isolation after an appropriate parameterization. Then, the stationary solution of the entire model is obtained as normalized product of the distributions of the sub-models. Although the most important example of product-form models is defined in terms of queueing networks, i.e., the BCMP theorem [Baskett et al. (1975)], the investigation of this property has involved almost all the other formalisms with an underlying CTMC. In the case of GSPNs a set of results are presented in [Coleman et al. (1996); Balbo et al. (2002)]. However, more recently, the problem of defining a unique framework for the product-form analysis of Markovian models has been investigated. In our opinion a major result is stated in [Harrison (2003)], where the author introduces the Reversed Compound Agent Theorem (RCAT) whose low-level formulation allows for its application despite of the formalism used to specify the interacting models. Using this result, and a generalization of the Markov implies Markov property [Muntz (1972)], in [Marin (2009)] we propose a unifying framework for combining product-form models defined in terms of different formalisms. In particular, we show how it is possible to model G-networks and multiclass queueing networks using GSPNs and then, we give conditions that allow the composition of these building-blocks such that the stationary solution is in product-form. This is useful to model complex systems in a framework where different types of sub-models can be combined maintaining the product-form solution. The various type of sub-models can be defined using different performance modeling formalisms, such as queueing networks and their extensions, GSPNs and some stochastic process algebra. The semantic of their combination is formally defined because the various sub-models are transformed in GSPNs with an equivalent underlying process. Moreover, we show how it is possible to compute the stationary distribution in product-form.

In this paper we present a novel approach to define a GSPN sub-model in product-form. In simple words, we aim to allow the modeler to store a library of product-form GSPN sub-models so that he can use them to describe complex architectures by specifying the way they cooperate. Note that we do not aim to define an automatic way to decide whether a sub-model is in product-form or not (although it can sometimes be done, e.g., [Coleman et al. (1996); Balbo et al. (2002)]), but we introduce the idea that given a library of models that are known to satisfy a set of properties, and a system described as a composition of these models, we can automatically decide whether that system has a product-form solution, and calculate it. According to this approach the GSPNs have to be appropriately annotated with some information that we shall describe in details in the following sections. It is worthwhile pointing out that within this framework it is possible to specify models such as G-queues, SPNs, multi-class queeing stations and PEPA models that interact. Moreover, we discuss a practical contribution of the theoretical result just illustrated above. Since the modularization and standardization is really important in this approach (e.g., we would like that the modeler may define a GSPN with his favorite tool and then just add the data needed for the product-form analysis) we propose the guidelines for an implementation of this framework based on the Petri Net Markup Language (PNML) [Weber and Kindler (2003)] and its extension to the modules [Kindler and Weber (2001)]. Finally, as further practical contribution, we show how to define GSPN models equivalent to G-queues.

We shall now recall the GSPN definition and the modular composition of GSPN submodels. We start from an example of modular combination of G-queues. Then we introduce the two basic properties for product-form models, i.e., the Markov implies Markov property and the Reversed Compound Agent Theorem, formulated at the CTMC level. We present the proposed framework to combine different sub-models into a unique GSPN that maintains the product-form solution. The submodels can be defined in terms of different formalisms and can be combined because thy can be transformed in GSPNs whose underlying process is equivalent. We discuss how to implement the framework by using PNML. Finally, we present some examples of application of the proposed technique.

## GSPN FORMALISM

In this section we briefly recall the Generalized Stochastic Petri Nets (GSPN) definition. A GSPN is a 8-tuple:

$$GSPN = (\mathcal{P}, \mathcal{T}, I(\cdot,\cdot), O(\cdot,\cdot), H(\cdot,\cdot), \Pi(\cdot), w(\cdot,\cdot), \mathbf{m_0})$$

where: $\mathcal{P} = \{P_1, \ldots, P_M\}$ is the set of $M$ places, $\mathcal{T} = \{t_1, \ldots, t_N\}$ is the set of $N$ transitions (both immediate and timed). $I(t_i, P_j) : \mathcal{T} \times \mathcal{P} \to \mathbb{N}$ is the input function, $1 \le i \le N$, $1 \le j \le M$, $O(t_i, P_j) : \mathcal{T} \times \mathcal{P} \to \mathbb{N}$ is the output function, $1 \le i \le N$, $1 \le j \le M$, $H(t_i, P_j) : \mathcal{T} \times \mathcal{P} \to \mathbb{N}$ is the inhibition function, $1 \le i \le N$, $1 \le j \le M$. $\Pi(t_i) : \mathcal{T} \to \mathbb{N}$ is a function that specifies the priority of transition $t_i$, $1 \le i \le N$, $\mathbf{m} \in \mathbb{N}^M$ denotes a marking or state of the net, where $m_i$ represents the number of tokens in place $P_i$, $1 \le i \le N$, $w(t_i, \mathbf{m}) : \mathcal{T} \times \mathbb{N}^M \to \mathbb{R}$ is the function which specifies for each timed transition $t_i$ and each marking $\mathbf{m}$ a state dependent firing rate, and for immediate transitions a state dependent weight, and finally $\mathbf{m_0} \in \mathbb{N}^M$ represents the initial state of the GSPN, i.e., the number of tokens in each place at the initial state. For each transition $t_i$ let us define the input vector $\mathbf{I}(t_i)$, the output vector $\mathbf{O}(t_i)$ and the inhibition vector $\mathbf{H}(t_i)$ as follows: $\mathbf{I}(t_i) = (i_1, \ldots, i_M)$, where $i_j = I(t_i, P_j)$, $\mathbf{O}(t_i) = (o_1, \ldots, o_M)$, where $o_j = O(t_i, P_j)$ and $\mathbf{H}(t_i) = (h_1, \ldots, h_M)$, where $h_j = H(t_i, P_j)$. Function $\Pi(t_i)$ associates a priority to transition $t_i$. If $\Pi(t_i) = 0$ then $t_i$ is a timed transition, i.e., it fires after an exponentially distributed firing time with mean $1/w(t_i, \mathbf{m})$, where $\mathbf{m}$ is the marking of the net. If $\Pi(t_i) > 0$ then $t_i$ is an immediate transition and its firing time is zero. We say that transition $t_a$ is enabled by marking $\mathbf{m}$ if $m_i \geq I(t_a, P_i)$ and $m_i < H(t_a, P_i)$ for $i = 1, \ldots, M$ and no other transition of higher priority is enabled. The firing of transition $t_i$ changes the state of the net from $\mathbf{m}$ to $\mathbf{m} - \mathbf{I}(t_i) + \mathbf{O}(t_i)$. The reachability set $RS(\mathbf{m_0})$ of the net is defined as the set of all markings that can be reached in zero or more firings from $\mathbf{m_0}$. We say that marking $\mathbf{m}$ is tangible if it enables only timed transitions and it is vanishing otherwise. For a vanishing marking $\mathbf{m}$ let $\mathcal{T}_\alpha$ be the set of enabled immediate transitions. Then the firing probability for any transition $t_i \in \mathcal{T}_\alpha$ and any state $\mathbf{m}$ is proportional to its weight. Given a tangible marking $\mathbf{m}$ the transition with the lowest associated stochastic time fires. A GSPN is represented by a graph with the following conventions: timed transitions are white filled boxes, immediate transitions are black filled boxes, places are circles, if $I(t_i, P_j) > 0$ we draw an arrow from $P_j$ to $t_i$ labelled with $I(t_i, P_j)$, if $O(t_i, P_j) > 0$ we draw an arrow from $t_i$ to $P_j$ labelled with $O(t_i, P_j)$, if $H(t_i, P_j) > 0$ we draw an circle ending line from $P_j$ to $t_i$ labelled with the value of $H(t_i, P_j)$, the marking $\mathbf{m}$ is represented by a set of $m_j$ filled circles representing the tokens in place $P_j$ for each $j = 1, \ldots, M$. For ordinary nets we do not use labels for the arrows. If we do not specify the weight of immediate transitions it is assumed to be 1 (usually we do this when we are sure there are no conflicts among immediate transitions). GSPN analysis consists in finding the steady-state probability for each tangible marking of the reachability set, from which one can derive other average performance indices. Some analysis techniques are presented in Marsan et al. (1995).

## GSPNs AND MODULES

The problem of giving a correct syntax and semantic of modular compositions of GSPNs has been addressed by several authors. In fact, the modularity allows for a definition of the models that is coherent with the principles of software and hardware engineering. In this paper, we use the module definition as proposed in [Kindler and Weber (2001)]. The main idea is that a module can be instantiated several times with possibly different parameterizations. It has an input and an output *interfaces* that allow the modeler to define how every instance interacts with the rest of the model, and an *internal implementation* that is invisible to the user. This approach could somehow be seen as the well-know procedure call schema implemented by most of the programming languages, where the input/output interfaces may be interpreted as the formal input/output parameters and the instances of a module as the procedure call. Within this interpretation, when the modeler connects the interfaces of a module with other elements of the model, he is defining the association between the actual and the formal parameters. In the following example we show a GSPN module whose underlying CTMC is equivalent to that of a G-queue, see [Gelenbe (1991)].

**Example 1 (GSPN model of a G-queue)** *G-queues are the smallest components of G-networks. They have been successfully used in a wide range of applications such as the analysis of database systems, communication networks or neural networks. In its simplest definition, a G-queue is a single-class queueing center with exponential service time distribution. Two arrival streams of customers are allowed: one for the so-called positive customers that exactly behave like ordinary customers in standard queueing stations, and the other for the negative customers. When one of these arrives to the station it can either delete a queued (positive) customer, if any is present, or simply vanish if the G-queue is empty. By now, we assume Poisson independent arrivals for positive and negative customers. Figure 1 illustrates a possible GSPN representation of a G-queue.*

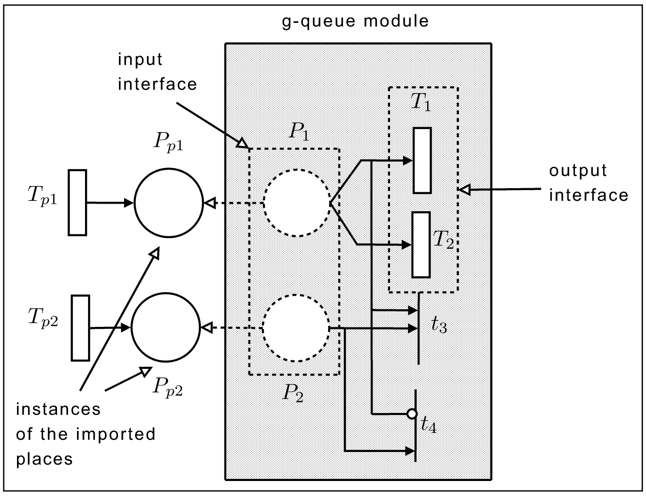


Figure 1: GSPN module equivalent to a G-queue.

*The module consists of two input places $P_1$ and $P_2$. The former stores a token for each positive customer in the station, while the latter stores one token at a negative customer arrival epoch. Notice that if there is one token in $P_2$ then either immediate transition $t_3$ or $t_4$ is* *enabled. The firing of $t_3$ consumes also a token from $P_1$ (positive customer deletion), while the firing of $t_4$ simply consumes the token in $P_2$ (i.e., the queue is empty and it vanishes). Moreover, it is immediate to observe that in every tangible marking of the net there are no tokens in $P_2$. Finally, $T_1$ and $T_2$ model the service of a customer. We use two transitions in order to straightforwardly model two different routings for customers served in such a station. For example $T_1$ may model the departure of a positive customer and $T_2$ the departure of a negative customer. Therefore, the service rate of the station is $\mu = w(T_1, \cdot) + w(T_2, )$. The input places are associated with places $P_{p1}$ and $P_{p2}$. $T_{p1}$ and $T_{p2}$ represent a hypothetical connection of a net with an instance of the module.*

**A brief introduction to concept of GSPN module.** In order to keep this paper self-contained, in this part we review the main concepts concerning the idea of modularization that we refer to. For a formal definition of the syntax and of the semantic we refer to [Kindler and Weber (2001)]. Note that other approaches to PN modularization are available in literature, e.g., that used by Timenet [Zimmermann et al. (2000)], but the passage from one to the other is not complicated.

Informally, we can say that a module is a net with an interface. We can create several instances of a module, but only the objects specified in its interface are accessible from outside the instance. What is not in the interface is called internal implementation. Referring to the Object Oriented Programming, this corresponds to the encapsulation feature. The interface consists of two parts: the input part (formed by the imported objects) and the output part (formed by the exported objects). Imported objects play the same role of formal parameters in the programming languages. Indeed, they are representatives of objects that are provided when the module is instantiated. Conversely, the objects that are exported are defined inside the implementation of a module (e.g., they may be provided as referred objects for an input interface of other module instances).

One can import and export three type of objects, i.e., places, transitions and symbols. The import and export of symbols allows us to define the parameterization of the modules. For instance, input symbols may be the transition rates, the number of tokens in a place of the internal implementation of the module and so on. The technique described in [Kindler and Weber (2001)] is really flexible, so one can just import or export functions, or anything else which can be useful for the modeler purposes. In the following we use input (output) object and imported (exported) object as synonymous.

Let us now reconsider the module defined in Example 1 and let us build a simple G-network using the GSPN modularization.

**Example 2 (G-networks)** *A composition of G-queues is called G-network. These models have shown to be suitable for the analysis of several software and hardware architectures. Let us consider the G-network depicted in Figure 2-(A). The G-network consists of two nodes, $C1$ and $C2$, with service rates* 3.0 *and* 2.0*, respectively. When customers leave $C1$ they can enter in $C2$ either as positive or negative customers, with probability* 0.9 *and* 0.1*. Customers may arrive from outside to $C2$ with rate $\lambda$. Once a customer is served by $C2$ it can leave the system with probability* 0.4 *or go back to $C1$ with probability* 0.6*.*

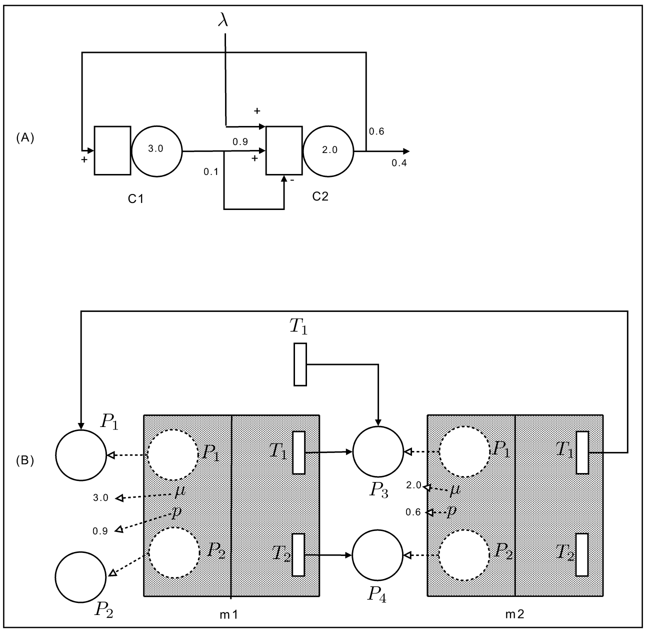


Figure 2: Use of GSPN modules to describe a G-network. (A) the original model. (B) the module composition.

*In order to give a GSPN representation of such a network we use a composition of two instances of the module introduced in Example 1. Actually, we added two symbols in the input interface, $\mu$ and $p$, which represent the service rate of the node, and the probability of firing of $T_1$ with respect to $T_2$. Therefore, $p$ and $\mu$ are used in the module definition to specify the rates of $T_1$ and $T_2$ in an obvious way: $w(T_1, \cdot) = p\mu$ and $w(T_2, \cdot) = (1-p)\mu$. Figure 2-(B) illustrates two instances of the module, $m1$ and $m2$, that are equivalent to the G-network of Figure 2-(A). In particular, the dotted arrows associate an object of an input interface with a concrete object (e.g., place $P_1$ in $m2$ with $P_3$, or $\mu$ in $m1$ with* 3.0*). Note that, since the* scope *of the object names is the module instance itself, the net has no conflicts on names, e.g., $P_1$ in instance $m1$ cannot be confused with $P_1$ of the net.*

## THE PRODUCT-FORM FRAMEWORK

In this section we present a framework to represent complex models combining different product-form submodels into a unique GSPN that maintains the product-form solution. This work is based on two results, i.e., the Reversed Compound Agent Theorem (RCAT) [Harrison (2003)] and the Markov implies Markov property ($M \Rightarrow M$) [Muntz (1972)]. After formally defining the composition rules of the module instances, we show that, although deciding whether a GSPN model satisfies $M \Rightarrow M$ or RCAT conditions is generally very difficult to do algorithmically, it is possible to store some information in the module descriptions that will allow a tool to automatically decide if a composition of such models has product-form solution and then derive the stationary distribution. As already mentioned, this means that the modeler works with a library of product-form models that have been opportunely annotated and that can be equivalent to G-queues, BCMP stations or other models that have been proved to be in product-form. GSPNs in product-form are studied in [Balbo et al. (2002)] and they are defined as GSPNs reducible to SPNs in Coleman, Henderson et al. product-form [Coleman et al. (1996)].

**RCAT and the $M \Rightarrow M$ property.** In this part we informally introduce RCAT and the $M \Rightarrow M$ property. Since the product-form analysis requires to study each components *as if* it were in isolation, we give the definition of what we mean by an isolated instance of a module (IIM).

**Definition 1 (Isolated instance of a module)** *Given an instance of a module in a net, its IIM is defined as follows:*

1. *For each input transition $T_i$ of the module we associate a transition with a null input vector and rate $\chi_{ti}$.*
2. *For each input place $P_j$ we associate a place which is fed by a transition $T_{pj}$ with a null input vector and an rate $\chi_{pj}$.*

*The rates $\chi_{ti}$ and $\chi_{pj}$ for each input transition $T_i$ and each input place $P_j$ are the* input rates *of the IIM, and $\mathcal{I}$ is the set of input rates.*

As an instance we can consider the net of Figure 1 where we can observe an IIM of the G-queue. The input rates are the rates of $T_{p1}$ and $T_{p2}$. We now introduce the set of reachable states of a module.

**Definition 2 (Reachability set of a module)** *The reachability set of a module is the set of all the markings reachable from its IIMs.*

Note that, in general, the reachability set of a module is not finite, and this is one of the reasons that makes the automatic decision of the following properties a very difficult task.

In order to simplify the formulation of RCAT and $M \Rightarrow M$ for GSPN modules, we limit the output objects to be transitions or symbols. This can be done without loss of generality possibly using immediate transitions.

**Definition 3 (RCAT-compatible IIM)** *We say that an IIM of a module is* RCAT-compatible *if and only if the following three conditions are satisfied:*

1. *For every tangible state, the instances of the input transitions must be always enabled. Informally, we can say that the module internal implementation cannot inhibit the input transition in any tangible marking.*
2. *Let $\mathbf{m}$ be a tangible marking of the reachability set. Then, if $T_o$ is an output transition there must exist one tangible marking $\mathbf{m}'$ such that $\mathbf{m}$ is reachable by $\mathbf{m}'$ through the firing of $T_o$.*
3. *For every pair of tangible marking $\mathbf{m}$ and $\mathbf{m}'$ such that $\mathbf{m}$ is reachable from $\mathbf{m}'$ through the firing of output transition $T_o$ the following relation holds:*

$$\pi(\mathbf{m}')w(T_o, \mathbf{m}') = K_o \pi(\mathbf{m}), \tag{1}$$

*where $\pi(\mathbf{m})$ is the stationary probability of marking $\mathbf{m}$ and $K_o \in \mathbf{R}^+$.*

These three conditions are just a rewriting of RCAT conditions [Harrison (2003)]. Finally, we observe that $K_o$ is a constant which is associated with each output transition $T_o$ that in general depends on the structure of the module and the input rates.

**Definition 4 ($M \Rightarrow M$-compatible module)** *We say that a module is $M \Rightarrow M$-compatible if and only if the following three conditions are satisfied:*

1. *See Condition 1 of RCAT-compatible definition.*
2. *See Condition 2 of RCAT-compatible definition.*
3. *Let $\mathbf{m}$ be a tangible state reachable from and $\mathcal{M} = \{\mathbf{m}'\}$ through the firing of an output transition $T_o$. Then, the following relation holds:*

$$\sum_{\mathbf{m}' \in \mathcal{M}} \pi(\mathbf{m}')w(T_o, \mathbf{m}') = K_o \pi(\mathbf{m}), \tag{2}$$

*where $\pi(\mathbf{m})$ is the stationary probability of marking $\mathbf{m}$ and $K_o$ is a linear combination of the input rates.*

In this case, it is not immediate to see that the conditions on the GSPN module are equivalent to the $M \Rightarrow M$ property. Indeed, this property is formulated in the context of queueing theory, therefore it involves concepts such as customers, class of customers, work-conserving an so on. The proof of the equivalence can be found in [Marin (2009)] and is based on a generalization of the $M \Rightarrow M$.

**Product-form composition and derivation of the stationary probabilities.** Let us introduce the problem of the product-form composition of the module instances with an example. Suppose that $m1$ and $m2$ are instances of RCAT-compatible module(s). Our aim is to define the appropriate input rates of the IIMs of $m1$ and $m2$ such that if $\mathbf{m} = (\mathbf{m}_1, \mathbf{m}_2)$ is a state of the original net, and $\mathbf{m}_1$ ($\mathbf{m}_2$) the state of $m1$ ($m2$), then:

$$\pi(\mathbf{m}) \propto \pi_1(\mathbf{m}_1)\pi_2(\mathbf{m}_2),$$

where $\pi(\mathbf{m})$, $\pi_1(\mathbf{m}_1)$ and $\pi_2(\mathbf{m}_2)$ are the stationary distributions of the whole net and of the IIMs of $m1$ and $m2$, respectively.

Obviously, these operations have not to be manually performed by the modeler, but we expect a tool to do them automatically. Indeed, the difficult task is the definition of the input rates.

Both $M \Rightarrow M$ and RCAT give a way to set these rates and they depend on the solution of a system called *traffic equation system.* Note that, in this framework it is not the case that the system of traffic equations is always linear as for example in BCMP queueing networks [Baskett et al. (1975)]. Therefore, we are able to study more general cases of product-form models than those based on the analysis of queueing networks.

The traffic equations depend on they way the module instances are connected. We allow two types of connections as specified by the following definition, where we consider that the arc weights are all 1.

**Definition 5 (Valid net)** *A net consisting of instances of $M \Rightarrow M$-compatible or RCAT-compatible modules is* valid *if the connection among the instances of the modules are such that for each instance:*

1. *if $T_i$ is an input transition then it is associated with just one output transition of another instance or a transition with null input and output vector and vice-versa*
2. *each place of the net is associated with one input place and an output transition which is not associated with an input transition can have an outgoing arc to just one place.*

In a valid net the interactions among the module instances are pairwise. In other words, at most two instances can change their markings as a consequence of the firing of a transition. Pairwise interactions are the only interactions that are considered both by RCAT and the $M \Rightarrow M$ property.

It is worthwhile pointing out that the validity of a net can be decided by a trivial algorithm. The implications of these definitions on the modelling of real systems are discussed in the original papers presenting the $M \Rightarrow M$ property and RCAT.

## THE FRAMEWORK IN PRACTICE

In this section we illustrate how we use the theoretical results recalled in the previous section to specify complex systems with product-form solutions. Informally, we can say that once the modeler has chosen the modules to instantiate, he/she connects them in one of the two ways that have been described. This operation can be seen as a graphical way to specify the traffic equations. We think that this is the key-point of our approach, i.e., the modeler uses a library of module whose behavior is known and when he connects them he is simply specifying the system of traffic equations. Note that, although this idea may seems trivial, it should be pointed out that it can be implemented thanks to the combination of the recent theoretical results about product-form such as RCAT and the idea of mapping each formalism into equivalent GSPNs.

What do we need to know about a module to be able to generate the system of traffic equations? As we already mentioned, the analysis of a single module with the aim of deciding whether it is RCAT-compatible or $M \Rightarrow M$-compatible may be a really hard task. Specifically, it can be often the case that the reachability set of the module is not finite. In order to overcome this problem, we introduce the concept of product-form GSPN module (PF-GSPN module). Let $\mathcal{I}$ be the set of the input rates and $\mathcal{V}$ be the set of the parameters of the module.

**Definition 6 (PF-GSPN module)** *A PF-GSPN module is a module with the following features:*

- *$f_{RCAT} : \mathcal{I} \times \mathcal{V} \to \{\boldsymbol{true}, \boldsymbol{false}\}$ is a boolean function which assumes the value **true** if, for a given parameterization, the module is RCAT-compatible*
- *$f_{M \Rightarrow M} : \mathcal{I} \times \mathcal{V} \to \{\boldsymbol{true}, \boldsymbol{false}\}$ which assumes the value **true** if, for a given parameterization, the module is $M \Rightarrow M$-compatible.*
- *For each output transition $T_o$, $K_o : \mathcal{I} \times \mathcal{V} \to \mathbf{R}^+$ is the function which specifies the reversed rate of $T_o$ in case of RCAT-compatibility or the sum of the reversed rates in case of $M \Rightarrow M$-compatibility. Obviously $K_o$ is defined only if $f_{RCAT}(\mathcal{I}, \mathcal{V}) \vee f_{M \Rightarrow M}(\mathcal{I}, \mathcal{V})$ is true.*

We now illustrate a set of example of PF-GSPN modules.

**Example 3 (G-queue)** *Let us consider again the G-queue of Example 1. In this case we have $\mathcal{I} = \{\chi_{p1}, \chi_{p2}\}$ and $\mathcal{V} = \{\mu, p\}$. The station is known to be always in RCAT product-form, [Harrison (2003)], while if fulfills* *the $M \Rightarrow M$ property only if there are not negative customer arrival (i.e., it is a standard exponential queue), therefore, we have:*

$$f_{RCAT}(\chi_{p1}, \chi_{p2}, \mu, p) = \textbf{\textit{true}} \quad \textit{always}$$
$$f_{M \Rightarrow M}(\chi_{p1}, \chi_{p2}, \mu, p) = \textbf{\textit{true}} \quad \textit{if } \chi_{p2} = 0$$

*From the G-network analysis [Gelenbe (1991)] the stationary probability of observing $m$ customers in $P_1$ is $(1-\rho)\rho^m$, where $\rho = \chi_{p1}/(\mu + \chi_{p2})$. Then, we straightforwardly obtain:*

$$\begin{aligned} K_1(\chi_{p1}, \chi_{p2}, \mu, p) &= \frac{\chi_{p1}\mu}{\chi_{p2}+\mu} p \\ K_2(\chi_{p1}, \chi_{p2}, \mu, p) &= \frac{\chi_{p1}\mu}{\chi_{p2}+\mu}(1-p) \end{aligned}$$

**Example 4 (A model of a shared bus contention)** *In this example we address the problem introduced in [Afshari et al. (1982)], where the authors propose a queueing model to study the access to a shared bus of a set of customers that are clustered into $R$ classes. The authors assume that the bus is able to handle $K$ simultaneous transmissions. As soon as a channel of the bus becomes available, a waiting customer is chosen with uniform probability among the queued ones to get served. The service time distribution is exponential and identically distributed for all the customer classes. In the paper the authors prove the stationary distribution and that the station satisfies the $M \Rightarrow M$ property.*

*In Figure 3 we propose a module of this system considering $R = 2$ classes of customers. Customers of class 1 and 2 arrive to input places $P_1$ and $P_2$, respectively. Place $P_5$ contains as many tokens as the free channels of the bus are. Immediate transitions $t_3$ and $t_4$ model the contention policy of the bus, i.e. their weight function is $w(t_3, \mathbf{m}) = m_1$ and $w(t_4, \mathbf{m}) = m_2$, where $\mathbf{m} = (m_1, \ldots, m_5)$ is a tangible marking and $m_i$ denotes the number of token in $P_i$. The rates of timed transition $T_1$ and $T_2$ are $\mu$. Finally, $K$ is the initial number of tokens in place $P_5$, i.e., the number of channels of the shared bus. In this case $\mathcal{I} = \{\chi_{p1}, \chi_{p2}\}$ and $\mathcal{V} = K, \mu$. As mentioned, in [Afshari et al. (1982)] the authors prove that the model satisfies the $M \Rightarrow M$ property, the $f_{M \Rightarrow M}(\mathcal{I}, \mathcal{V}) = \textbf{\textit{true}}$ always. In [Marin (2009)] we prove that it satisfies RCAT conditions if $K = 1$, i.e., $f_{RCAT}(\mathcal{I}, \mathcal{V}) = \textbf{\textit{true}}$ if $K = 1$. Finally, $K_1(\mathcal{I}, \mathcal{V}) = \chi_{p1}$ and $K_2(\mathcal{I}, \mathcal{V}) = \chi_{p2}$.*

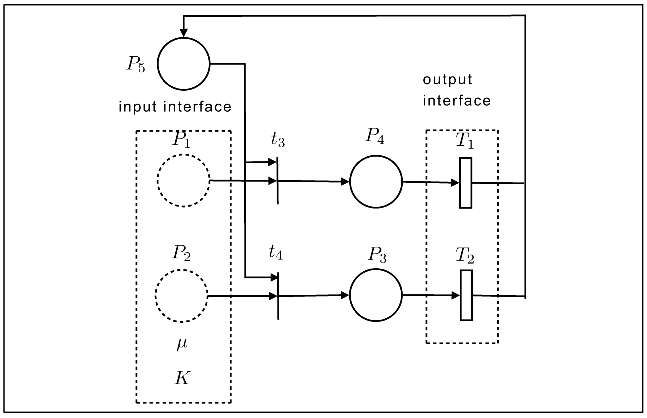


Figure 3: PF-GSPN module of the queueing model of a shared bus contention for two classes of customers.

**Automatic derivation of the traffic equations.** In order to be able to decide whether a valid net is in product-form, and in this case provide the stationary solution, we need to generate and solve the set of traffic equations. The unknowns of these equations are the input rates of the module instances of the net. If we are able to solve the traffic equations we can check if they satisfy the conditions for the $M \Rightarrow M$ or RCAT application for each instance of the module using functions $f_{\text{RCAT}}$ and $f_{M \Rightarrow M}$. If this is the case, then we can derive the stationary distribution of the IIMs associated with every module instance using the input rates obtained by the solution of the traffic equations. Then, the stationary solution of the original net is proportional to the product of these stationary solutions.

In the following, in order to avoid conflicts of names in the equations, we use the notation instance_name.object (e.g., $m1.\chi_{p1}$). A valid net admits only two types of connections. Each of these generate the following equations:

- Suppose that output transition $T_o$ of instance $mk$ is the input transition $T_i$ of instance $mh$ with $mk \neq mh$. In this case we have $mh.\chi_{ti} = mk.K_o(mk.\mathcal{I}, mk.\mathcal{V})$.
- Suppose that the set output transitions $\mathcal{T}^* = \{mk.T_o\}$ is such that all the elements $mk.T_o$ have an outgoing arc to $P_{Ii}$ that is an instance of input place $P_i$ of $mh$, with $mk \neq mh$. In this case we have that:

$$mh.\chi_{ti} = \sum_{mk.T_o \in \mathcal{T}^*} mk.K_o(mk.\mathcal{I}, mk.\mathcal{V})$$

It is out of the scope of this paper to address the problem of an efficient solution of such a system. However, using Muntz's result [Muntz (1972)] we can state that the system is linear if all the instances of the modules are $M \Rightarrow M$-compatible. An approach used in [Argent-Katwala (2006)] is to export the equations in ASCII format and solve them using general purpose software on Mathematics. If all the used modules have a finite reachability set, then the algorithm presented in [Marin and Rota Bulò (2009)] may be used.

## EXAMPLE

The purpose of the following example is to show how the technique previously described may be applied to study a system consisting of sub-components that cannot be modeled by ordinary queueing stations.

**System description.** Two classes of requests arrive through a communication line from two networks. The communication channel is bidirectional and has a waiting room where the packets are stored. When a transmission is completed a packet is chosen from the waiting room according to a random policy. Once transmitted, the requests are pre-processed by an ad-hoc system and finally sent to the database. Some of the requests of the first class may be converted into requests of the second class. In some cases, the pre-processing phase may decide to cancel a transaction that has already been sent to the database. Some transactions fail, and have to be sent back to the communication line to get processed again. The database answers are sent back to the clients through the channel. Figure 4 shows a sketch of this system.

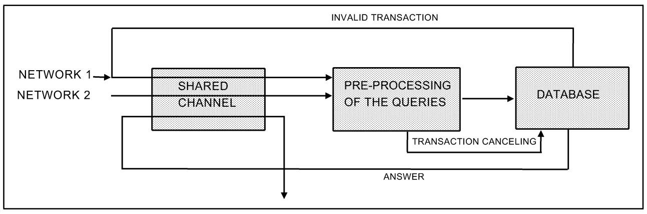


Figure 4: Software architecture analyzed in the Example section.

**Model description** The modeling assumptions are the following. The channel behaves like a shared bus as described in Example 4 and the database is modeled by a G-queue as described in Example 1. This means that the service time distributions are exponential and class independent both for the database and the communication lines. The pre-processing of the requests is modeled by the Module of Figure 5. For brevity, we do not specify each phase of the processing, but it is important to note that fork-join constructs are present, and this makes the model impossible to be studied by most of the existing product-form analyzer. For this module, we have $\mathcal{I} = \{\chi_{p1}, \chi_{p2}\}$ and $\mathcal{V} = \emptyset$. In [Marin (2009)] is proved that $f_{M \Rightarrow M}(\mathcal{I}) = f_{\text{RCAT}}(\mathcal{I}) = \textbf{true}$, and $K_4(\mathcal{I}) = \chi_{p1}$ and $K_5(\mathcal{I}) = \chi_{p2}$ (note that also the stationary distribution is provided). Another module that we use and that has not been previously described is a switching module. This simply routes the tokens that arrive to its incoming place according to a static probability as depicted by Figure 6. In our framework we can model the system as depicted by Figure 7, where $m1$ is an instance of the module of a shared bus described in Example 4, $m2$ is an instance of a router, $m3$ an instance of the SPN module of Figure 5 and, finally, $m4$ is an instance of a G-queue module described in Example 1. The model parameters are: the firing rates of $T_A$ ($\lambda_A$) and $T_B$ ($\lambda_B$), i.e., the arrival rates of the requests, $\mu_{\text{TR}}$, i.e., the transmission rate of one line of the channel, $p_{\text{SWITCH}}$, i.e., the probability that a request of the first type becomes a request of the second type, $p_{\text{ERR}}$, i.e., the probability of reprocessing of a transaction and finally $\mu_{\text{SER}}$, i.e., the service rate of the database.

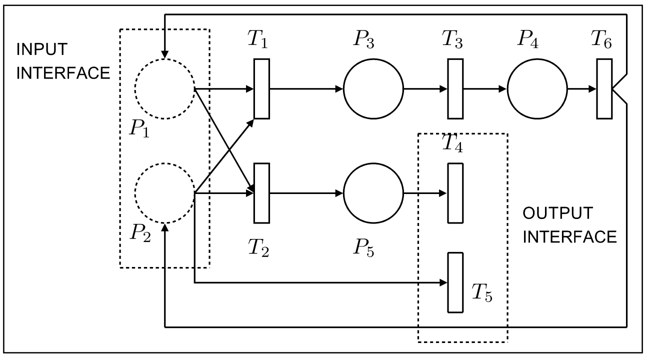


Figure 5: SPN module of the query pre-processing phase.

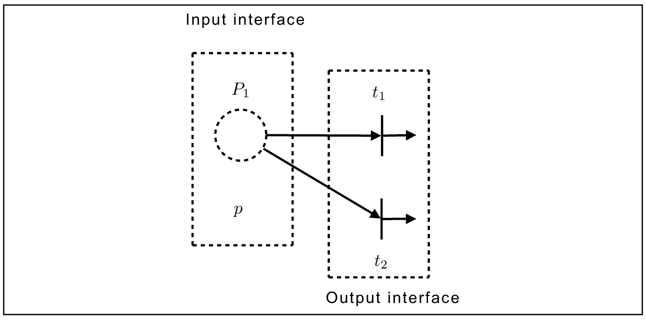


Figure 6: Simple router module. We have $w(t_1, \cdot) = p$ and $w(t_2, \cdot) = 1 - p$. $\mathcal{I} = \{\chi_{p1}\}$ and $\mathcal{V} = \{p\}$. $f_{M \Rightarrow M} = f_{\text{RCAT}} = \textbf{true}$ and $K_1(\mathcal{I}, \mathcal{V}) = p\chi_{p1}$, $K_2(\mathcal{I}, \mathcal{V}) = (1 - p)\chi_{p1}$.

**Traffic equations.** In our framework we can algorithmically derive the traffic equations using the rules presented in the previous sections, obtaining:

$$\begin{cases} m1.\chi_{p1} = \lambda_A \\ m1.\chi_{p2} = \lambda_B \\ m1.\chi_{p3} = m4.K_2 = \frac{m4.\chi_{p1}}{m4.\chi_{p2} + \mu_{\text{SER}}}(1 - p_{\text{ERR}})\mu_{\text{ERR}} \\ m2.\chi_{p1} = m1.K_1 = m1.\chi_{p1} \\ m3.\chi_{p1} = m2.K_1 = m2.\chi_{p1} p_{\text{switch}} \\ m3.\chi_{p2} = m2.K_1 + m1.K_2 \\ \qquad = m2.\chi_{p1}(1 - p_{\text{switch}}) + m1.\chi_{p2} \\ m4.\chi_{p1} = m3.K_1 = m3.\chi_{p1} \\ m4.\chi_{p2} = m3.K_2 = m3.\chi_{p2} \end{cases}$$

Once derived the solution for the traffic equations, this is used to set the input rates of the IIMs of the module instances. Then, we observe that all the IIMs are RCAT-compatible and therefore the model is in product-form.

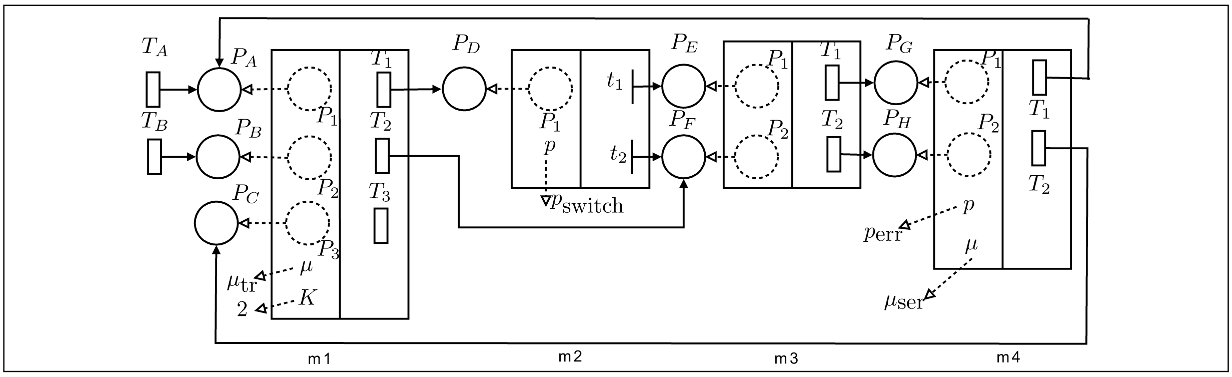


Figure 7: Modular composition of the system of Figure 4.

The stationary solution $\pi_i$ of each IIM for $i = 1, \ldots, 4$ is then derived and the stationary probabilities $\pi$ of the whole model are such that $\pi \propto \prod_{i=1}^{4} \pi_i$. Knowing the stationary distribution $\pi$ of the model allows us to compute some interesting performance indices, e.g., the mean response time of the database, or distribution of the number of customers in the communication line.

## CONCLUSION

In this paper we have presented a novel approach to analyze product-form GSPNs. Its main strengths are the high modularity and the fact it is capable to deal with several product-form model classes, such as BCMP queueing networks, G-queues, product-form SPNs, and so on. The idea underlying this work is to use the module concept as defined in [Kindler and Weber (2001)] to define product-form models. These have to be annotated in order to allow a software tool to take advantage from the product-form property in the analysis phase. It can be shown that all this work may be implemented using PNML without violating the standard. Finally, it is worthwhile pointing out that in this framework, a modeler is not supposed to have particular knowledge about product-form models or GSPN modeling. Indeed, modelers just need to pick some modules from a library and then create and connect their instances according to the simple rules that we have described. Then, the steady state analysis and the derivation of the desired performance indices can be automatically computed. Further research efforts should have two directions. One is the implementation or the extension of an existing tool capable to perform such an analysis. This should not be hard, since it suffices to specify an appropriate PNML grammar and use a symbolic tool to solve the traffic equations. Another research open problem could deal with the possibility of connecting the module instances with arcs with arbitrary weights. This introduces some complex problems in the analysis but would enhance the flexibility of the framework.

## REFERENCES

Afshari P.V.; Bruell S.C.; and Kain R.Y., 1982. *Modeling a new technique for accessing shared buses.* In *Proc. of the Computer Network Performance Symp.* ACM Press, New York, NY, USA, 4–13.

Argent-Katwala A., 2006. *Automated product-forms with Meercat.* In *SMCtools '06: Proc. from the 2006 workshop on Tools for solving structured Markov chains.* ACM, New York, NY, USA, 10.

Balbo G.; Bruell S.C.; and Sereno M., 2002. *Product Form Solution for Generalized Stochastic Petri Nets. IEEE Trans on Software Eng*, 28, no. 10, 915–932.

Baskett F.; Chandy K.M.; Muntz R.R.; and Palacios F.G., 1975. *Open, Closed, and Mixed Networks of Queues with Different Classes of Customers. J ACM*, 22, no. 2, 248–260.

Coleman J.L.; Henderson W.; and Taylor P.G., 1996. *Product form equilibrium distributions and a convolution algorithm for Stochastic Petri nets. Perform Eval, Elsevier*, 26, no. 3, 159–180.

Gelenbe E., 1991. *Product form networks with negative and positive customers. Journal of Applied Prob*, 28, no. 3, 656–663.

Harrison P.G., 2003. *Turning back time in Markovian process algebra. Theoretical Computer Science*, 290, no. 3, 1947–1986.

Kindler E. and Weber M., 2001. *A universal module Concept for Petri nets.* In G.J. und Robert Lorenz (Ed.), *Proc. of 8th Workshops AWPN.* Katholische Universität Eichstätt, Germany, 7–12.

Marin A., 2009. *On the relations among product-form stochastic models.* Ph.D. thesis, Università Ca' Foscari di Venezia, Venice.

Marin A. and Rota Bulò S., 2009. *A general algorithm to compute the steady-state solution of cooperating Markov chains.* In *Proc. of 17th Annual Meeting of the IEEE Inter. Symp. MASCOTS.* London, UK, 515–524.

Marsan M.A.; Balbo G.; Conte G.; Donatelli S.; and Franceschinis G., 1995. *Modelling with generalized stochastic Petri nets.* Wiley, New York, NY, USA.

Muntz R.R., 1972. *Poisson Departure Processes and Queueing Networks.* Tech. Rep. IBM Research Report RC4145, Yorktown Heights, New York.

Weber M. and Kindler E., 2003. *Petri Net Technology for Communication-Based Systems*, H. Ehrig, W. Reisig, G. Rozenberg, H. Weber ed., chap. The Petri Net Markup Language. 124–144.

Zimmermann A.; Freiheit J.; German R.; and Hommel G., 2000. *Petri Net Modelling and Performability Evaluation with TimeNET 3.0.* In *TOOLS '00: Proc. of the 11th Int. Conf. on Computer Perf. Eval.: Modelling Techniques and Tools.* Springer-Verlag, London, UK, 188–202.

]

# ENHANCING DISCRETE SIMULATION EXECUTIVE WITH SIMPLE CONTINUOUS SIMULATION AND ANIMATION SUPPORT

Norbert Adamko
Faculty of Management Science and Informatics
University of Zilina
Slovak Republic
Norbert.Adamko@fri.uniza.sk

## KEYWORDS

Combined discrete continuous simulation, run-time animation, simulation executive.

## ABSTRACT

The main topic of the paper is the enhancement of originally purely discrete simulation executive that is part of the ABAsim architecture with the support for continuous simulation and run-time animation. The resulting combined simulation executive utilises modular design that enables to optionally employ only requested parts of the executive. In contrast to the discrete simulation module, the continuous simulation and animation modules are utilising activity scanning technique to execute and synchronise their activities. Some applications that use the proposed combined simulation executive design are also discussed.

## INTRODUCTION

With the increasing need for simulation models that are able to reflect the reality (or thoughts) more precisely, the demand for continuous simulation support in simulation executives often arises. Even simulation models that have been designed to be purely discrete often need to be later enhanced with continuous, and thus more detailed, modelling of some activities (e.g. movement of vehicles). Since by the primary design of such models no continuous modelling was considered, the models are often based on proprietary architectures and discrete-only simulation executives, which are generally quite easy to implement but have limited modelling abilities. Due to a variety of reasons, the complete redesign of such models and utilisation of different simulation executive (or even architecture) might not always be feasible. Therefore, in order to be able to utilise existing parts of the model, the existing simulation executive has to be enhanced with the support for continuous simulation.

Nowadays, the animated graphical output has become an integral part of many simulation programs and tools and is often requested by clients of simulation studies. The animated graphical presentation of simulated activities during simulation run, so called *run-time animation*, provides simulation study clients as well as simulationists with clear and understandable indication of simulation model's state (supporting model validation) and the possibility to immediately interact with the simulation model. In order to support run-time animation, the simulation executive has to provide means for the synchronised execution of *animation activities* that realise the animation of simulation model activities, regardless of their type. For example, even discrete activity that models the movement of a vehicle (i.e. the vehicle location attribute changes discretely at the activity end) has to be presented to the user as smooth vehicle movement (utilising average speed of the vehicle). The integration of the animation support into the simulation executive can ease the implementation of run-time animation in simulation models, especially discrete ones. However, the employment of the animation should be optional and the additional overhead caused by the integration of the animation into the executive should be kept as low as possible.

On an example of the simulation executive that belongs to the *ABAsim architecture* (Adamko 2004), this paper will present one of possible approaches to the creation of combined simulation executive by enhancing original discrete simulation executive with simple optional support for continuous simulation and run-time animation.

## COMBINED SIMULATOR DESIGN

To guarantee the flexibility (e.g. optional execution) without changing the interface of the existing discrete simulator, and keeping the overhead as low as possible at the same time, modular design of the combined simulator has been proposed.

Two simulation modules create the base of the combined simulator – *discrete simulation module* (existing) and *continuous simulation module*, each responsible for the execution of respective types of activities. These modules are complemented by the *animation module*, which is responsible for the animation of simulated activities (and their presentation on the computer display), and *interface module* that intermediates the communication between simulation modules and the animation module.

## DISCRETE SIMULATION MODULE

Due to the fact that the proposed combined simulator is an enhancement of existing discrete simulation executive, the discrete simulation module (DSM) will govern the simulation and become the central controlling module of the combined simulator. The central module permits other modules to run by granting time quanta and for this limited time period hands the control of the simulation run over to them.

Prior to the description of discrete simulation module's operation, let us first explain how discrete activities and processes are modelled. The process is a sequence of naturally adjacent activities that together create a logical unit. Each process is started by delivering a *Start* message to it. During the processing of the *Start* message, the process starts its first (discrete) activity by planning a *Hold* message that models the end point of the activity (Fig. 1). Since the duration of discrete activity is generally known by its start, the time stamp of the *Hold* message can be set to the respective value (current simulation time value plus the duration of the activity). The delivery of this Hold message symbolises the end of the activity and all respective state changes bound with the activity are realised at this time. After the message has been handled, the process can start its next activity or terminate own execution (if the just finished was the last activity of the process) by sending the *Finish* message to its controlling entity (in ABAsim architecture it is an agent).

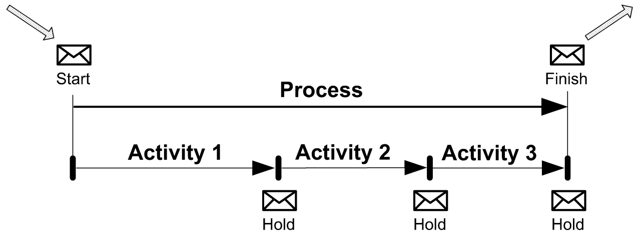


Figure 1: Process composed of three activities

The discrete simulation module consists of discrete simulation kernel and a future event list (FEL) that holds messages to be delivered at times specified by their time-stamp information. The original discrete simulation kernel utilised standard event scanning technique to manage the simulation time and deliver messages – during each simulation loop, a message with lowest time stamp was removed from the future event list, the simulation time was updated to the time-stamp value and the message was delivered to the addressee for processing.

The new discrete simulation kernel is modified and works in the following way. Assuming that all messages that have to be delivered at the current simulation time ($\boldsymbol{t_D}$) are already delivered and processed, the discrete simulation module can identify the time quantum $\boldsymbol{\Delta t_1}$ (Fig. 2) that is equal to the time difference of current simulation time $\boldsymbol{t_D}$ and the time stamp $\boldsymbol{t_V}$ of first future message planned for delivery ($\boldsymbol{\Delta t_1 = t_V - t_D}$). If this quantum is greater than zero, i.e. $\boldsymbol{\Delta t_1} > 0$, then the simulation run control is handed over to the continuous simulation module with the time grant for $\boldsymbol{\Delta t_1}$ to execute its tasks. Notice, that the continuous simulation is active only during idle times of the discrete simulation module. After the CSM finishes, the simulation run control is returned back to the discrete simulation module that identifies the time difference $\boldsymbol{\Delta t_2}$ between current simulation time $\boldsymbol{t_D}$ (notice that the simulation time of DSM is used) and the time stamp $\boldsymbol{t_E}$ of first future message planned for delivery ($\boldsymbol{\Delta t_2 = t_E - t_D}$). If the CSM fully consumed the granted time quantum (no message with time stamp $\boldsymbol{t_E < t_D + \Delta t_1}$ has been planned by this module), then $\boldsymbol{\Delta t_2 = \Delta t_1}$ and $\boldsymbol{t_E \equiv t_V}$.

Identified time quantum $\boldsymbol{\Delta t_2}$ is granted to the animator module and the simulation control is handed over to it. The animation module will execute "ex-post" animation for the simulation time interval <$\boldsymbol{t_D}$, $\boldsymbol{t_E}$).

After return of the control the DSM initiates flushing of the memory buffer (the records are either moved to the external disk memory for future use or are just deleted) and continues its operation (removing and processing the message with lowest time stamp and continuing with next simulation loop).

## CONTINUOUS SIMULATION MODULE

The continuous simulation module (CSM) is responsible for the simulation of continuous activities of processes.

In order to start the execution of continuous activity, the process must first register the continuous activity with the continuous simulation module. The registration is initiated from the discrete part of the model during the processing of *Start* or *Hold* messages of the respective process – if the next activity is a continuous one, the process (instead of sending a *Hold* message as it is done by discrete activity modelling) registers this activity with the CSM, providing, among other parameters, a *Hold* message that is to be delivered after the activity finishes. Since the duration of continuous activities is generally not known in advance (it is determined by the continuous simulation computation during run-time) the *Hold* message cannot be planned for delivery at specified time but will be delivered to the process after the activity identifies that its finishing condition has been met.

During the registration, the CSM includes the activity to the list of registered continuous activities that are ready for execution. All registered activities (each process can register only one continuous activity at a time) are then executed by the *continuous simulation kernel* (during its active time) utilising the periodic activity scanning approach with scanning period $\boldsymbol{\tau}^{\mathrm{C}}$. Besides the simple fixed step numerical integration methods, the application of variable step numerical integration methods is also possible; however this topic exceeds the scope of the paper.

Continuous simulation kernel is activated as described in previous chapter at current simulation time $\boldsymbol{t_D}$ (Fig. 2) and it is granted a time quantum $\boldsymbol{\Delta t_1}$. Granted amount of time $\boldsymbol{\Delta t_1}$ is fully consumed only if no message was planned (to be delivered on time $\boldsymbol{t_E}$; $\boldsymbol{t_E < t_D + \Delta t_1}$) during the CSM processing (as a result of continuous activity execution). If a message was sent to be delivered at the time $\boldsymbol{t_E}$, then only part $\boldsymbol{\Delta t_2}$ of granted time quantum $\boldsymbol{\Delta t_1}$ is used by CSM, the reason for this is that the CSM can only be active when there are no messages delivered or executed in discrete part of the model (DSM).

After the activation of CSM (the execution control was transferred from discrete simulation module), the local simulation time of continuous simulation module ($\boldsymbol{t_C}$) is set to the value $\boldsymbol{t_C} = \boldsymbol{t_D}$ and **utilisable time quantum $\boldsymbol{\Delta t_2}$** is initialised to the value of granted amount of time $\boldsymbol{\Delta t_2} = \boldsymbol{\Delta t_1}$.

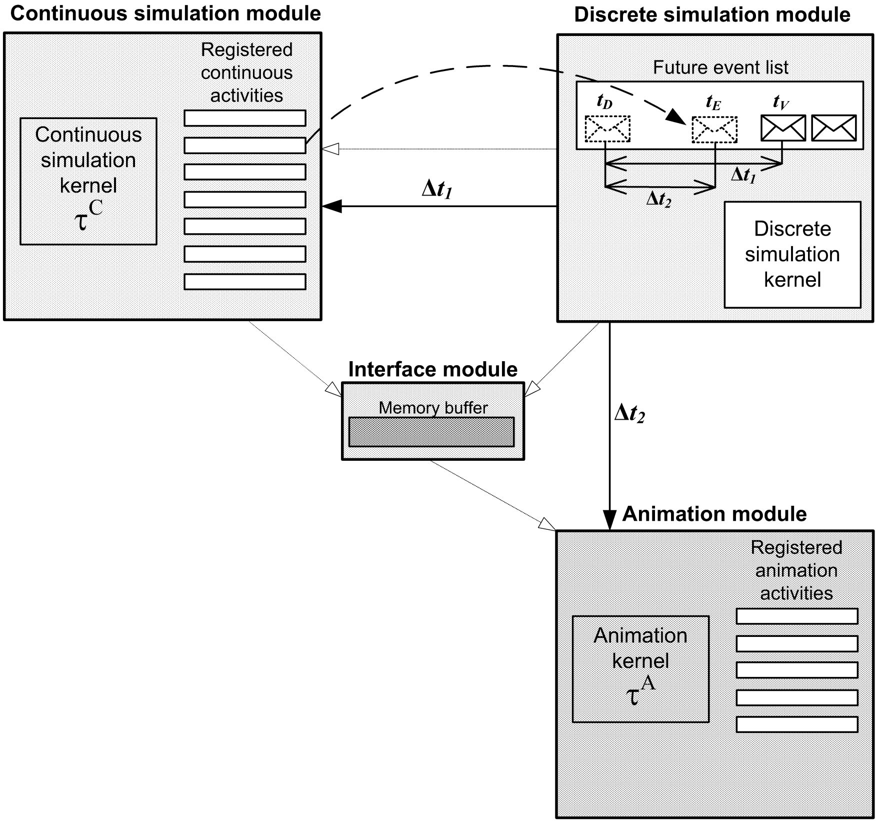

Continuous simulation module
Registered continuous activities
Continuous simulation kernel
$\tau^C$
Discrete simulation module
Future event list
$t_D$
$t_E$
$t_V$
$\Delta t_1$
$\Delta t_2$
$\Delta t_1$
Discrete simulation kernel
Interface module
Memory buffer
$\Delta t_2$
Animation module
Registered animation activities
Animation kernel
$\tau^A$

Figure 2: Combined simulation executive structure

If there are no continuous activities registered, the execution control returns immediately back to the discrete simulation module. If the continuous activity list is not empty, the CSM starts its execution – utilising classic activity scanning approach the registered activities are periodically scanned (usually involving some differential equation calculations) with scanning period equal to $\boldsymbol{\tau}^C$. Before each scan, the local simulation time of CSM is incremented by $\boldsymbol{\tau}^C$.

If the CSM realises that during the activity scanning a message with time stamp $\boldsymbol{t_E} < \boldsymbol{t_D} + \boldsymbol{\Delta t_2}$ was sent, the time quantum $\boldsymbol{\Delta t_2}$ is set to the value $\boldsymbol{\Delta t_2} = t_E - t_D$. There are, in general, two possible reasons for the message to be sent:

- During the activity scanning a message activation condition was triggered or a situation that influences the discrete part of the model (e.g. threshold variable value was reached) occurred.
- One of registered activities finished its execution, causing a *Hold* message to be send to the registering process (with time stamp $\boldsymbol{t_E}=\boldsymbol{t_C}$) and the exclusion of the activity from the continuous activity list. Notice, that continuous activity is finished by the delivery of the *Hold* message to the process, i.e. by exactly the same way as the discrete activities. The main difference being that by discrete activities, the delivery time of the *Hold* message is known by the activity start, whereas by continuous activities the delivery time is a result of computations handled by CSM.

At the end of the second phase of the activity scanning, the CSM tests whether the finishing condition $\boldsymbol{t_C} - \boldsymbol{t_D} = \boldsymbol{\Delta t_2}$ is met, i.e. the utilisable time quantum has been consumed. If the condition is met, the simulation control is returned back to the DSM, otherwise the CSM continues with next scanning period.

## INTERFACE MODULE

The interface module separates the simulation model from its graphical output on the computer screen. This module incorporates a memory buffer that is filled with information from discrete and continuous activities executed in respective modules. Activities that request animation chronologically store all required data for the graphical output in the buffer memory. Animation kernel then (after its activation) reads this data and independently (although under the direction of DSM) performs animation calculations and the graphical output.

The data stored in memory buffer can be saved to an external memory (during the flush operation initiated by DSM) for later realisation of post-run animation with proprietary solutions or utilising third party tools, e.g. Proof Animation (Henriksen 2000) or Animation Toolbox (ISL 2007).

## ANIMATION MODULE

The main task of the animation module (AM) is to execute registered animation activities resulting in the graphical presentation of simulation computation processes. *Animation activity* is a program routine (function) that performs the animation of given object by changing its attributes (e.g. position, colour, etc.) and optionally presenting its graphical representation on the computer screen.

The animation module is composed of the *animation kernel* that controls the animation and the *list of registered animation activities* that holds all currently active registered activities. Similarly to the continuous simulation module, the animation module is activated by the discrete simulation module kernel and it is granted the time quantum $\mathbf{\Delta t_2}$. At the time of the AM activation, all actions (parts of activities) that should happen during time interval $<\mathbf{t_D}, \mathbf{t_E})$ are already executed and the memory buffer of interface module contains animation data that were place here during this time interval by the discrete and/or continuous activities from respective modules.

Animation kernel first processes all relevant records from the interface module's memory buffer that could be put there by any discrete or continuous activity of currently running processes. The records contain at least information about the *code* of the requested animation activity and the *duration* of the animation activity, which are usually accompanied by additional data regarding the animation (e.g. object to be animated, movement speed, animation colour, etc.). Based on the code, requested animation activity is created, initialised with additional parameters provided and registered with the module by adding it to the list of currently active registered animation activities. Special attention has to be paid to the registration of animation activities that originate in continuous simulation module – the memory buffer usually contains more records that are related to the same animation activity (one for each scanning period of CSM). Therefore these records have to be joined into single compound animation activity.

After all records have been processed, the animation kernel (similarly to the continuous simulation module) periodically scans the animation activities with scanning period $\boldsymbol{\tau}^{\mathbf{A}}$, called the *animation step*. During the evaluation phase of the activity scanning the animation activity executes the respective animation tasks (e.g. moves the vehicle on screen by the distance depending on the vehicle speed and duration of the scanning period). If any animation activity finishes, it is automatically removed from the list and its instance is destroyed. After consuming the granted time $\mathbf{\Delta t_2}$, the animation module interrupts its execution and returns simulation control back to the discrete simulation module.

Let us present the basic animation principles on a very simple example. Typical animation activity is the movement activity that animates changes in position of an object. Imagine that we simulate the process of car movement utilising discrete simulation activity and we want to move the car from point A to point B over the distance of 200 m with an average speed of 5 m.s$^{-1}$. When the movement process is started (by a *Start* message) the time needed for the car to pass the distance is calculated (40 s in this case) and an appropriate *Hold* message is planned for delivery at the specified time. During the processing of this message, the car will change its position from the place A to the place B and the process finishes. To animate this simple movement, the process has to additionally put a record into the buffer of interface module to instruct the animation module about the movement animation request and to provide parameters for the animation (e.g. the object to be moved, the activity duration and the length of the movement) – this is done during the start of the process, i.e. before the process actually finishes (notice, however, that the finish time of the process is already known and planned). The animation module processes this record and creates respective animation activity instance, initialises its parameters and puts it into the list of active activities. The implementation of the animation activity is quite straightforward – the activity contains the animation method that is invoked by each scan of the animation activity (i.e. every $\boldsymbol{\tau}^{\mathbf{A}}$ time units). This method simply changes the position of the object by the distance that correspond to the movement during $\boldsymbol{\tau}^{\mathbf{A}}$ time units, in our example if the scanning period of animation module would be set to 0.1s, the animation activity would change the position of the car by 0.5 metres (the overall duration of the activity is 40 seconds and the car should move over 200 m in total, this means that during 0.1 second the car moves (0.1/40)*200 metres). The change in position can be immediately reflected on the computer screen (this approach is used by GDI drawing) or the screen output can be realised independently in asynchronous manner (typically used by DirectX or OpenGL visualisation).

The size of the scanning period $\boldsymbol{\tau}^{\mathbf{A}}$ of the AM controls the smoothness of the animation (this is very similar to the number of frames per second in a film) and indirectly also the speed of the simulation run – since the module execution is synchronised, the simulation modules cannot execute their activities before animation module finishes its tasks.

## SYNCHRONISATION ALGORITHM OF THE COMBINED SIMULATION EXECUTIVE

The synchronisation algorithm of the combined simulation executive with run-time animation is summarized in following table.

| Module | Step | Task | Conditions |
|---|---|---|---|
| DSM | 0 | Initialise the simulation time $t_S$ ($t_S = 0$) | |
| | 1 | End simulation run | FEL is empty or the simulation time limit has been reached |
| | 2 | Take out first event from FEL | |
| | 3 | Update the simulation time ($t_S = t_U$) | |
| | 4 | Process the event | |
| | 5 | Identify and grant time quantum $\mathbf{\Delta t_1}$ | |
| CSM | 6 | Apply periodic time activity scanning (with scanning period $\tau^C$ ) on all registered continuous activities | $\Delta t_1 \neq 0$ and the number of registered continuous activities $n_C \neq 0$ |
| DSM | 7 | Identify and grant time quantum $\mathbf{\Delta t_2}$ | |
| AM | 8 | Process buffer records and register respective animation activities | $\Delta t_2 \neq 0$ |
| | 9 | Apply periodic time activity scanning (with scanning period $\tau^A$) on all registered animation activities till the $\mathbf{\Delta t_2}$ is consumed. | $\Delta t_2 \neq 0$ and the number of registered animation activities $n_A \neq 0$ |
| DSM | 10 | Initiate the flushing of buffer memory | |
| IM | 11 | Flush the buffer memory | Buffer is not empty |
| DSM | 12 | Go to **step 1** | |

Table 1. The control algorithm of the combined simulation with run-time animation

## CONCLUSIONS

The presented combined simulation executive has been implemented as a part of the ABAsim agent based simulation architecture support libraries. Besides its utilisation in teaching process at the University of Zilina, the combined simulation executive (and the ABAsim architecture) has been used in large scale simulation models of transportation logistic systems, e.g. Villon (Adamko 2007). The experience with the simulation executive indicates that this solution provides efficient means for enhancement of existing, as well as newly designed, simulation models with simple continuous simulation and run-time animation. The chosen modular approach guarantees flexibility in optional employment of modules while keeping the simulation executive overhead low (if the additional modules are not utilised).

The proposed combined simulation executive design provides a simple and straightforward solution to the problem of the integration of continuous simulation and animation into existing simulation models and can be easily adapted to any discrete simulation executive that provides indication of the future event time stamp.

## ACKNOWLEDGEMENT

*This work has been supported by the Slovak Grant Foundation under grant No. 1/4057/07 "Agent oriented models of service systems" and by the National Research Program of Czech Republic under project No. MSM 0021627505 "Theory of Transportation Systems".*

## REFERENCES

Adamko, N., Klima, V., Kavička, A., Lekýr, M. 2004. "Flexible hierarchical architecture of simulation models". In *Proceedings of European simulation and modelling conference 2004*, Eurosis, Paris, 2004, pp.30-34, ISBN 90-77381-14-7.

Adamko, N., Klima, V., Kavička, A. 2007. "Villon – agent based generic simulation model of transportation logistic terminals", In *Proceedings of European simulation and modelling conference 2007*, Eurosis, St. Julian's, Malta, 2007, pp. 364-368, ISBN 978-90-77381-36-6

ISL. 2007. "Animations-Toolbox", *http://www.isl.org/products_ services/animation/index.php?lang=en*

Henriksen, J. O. 2000. "Adding animation to a simulation using Proof", In *Proceedings of the Winter simulation conference,* SCS International, 2000, pp. 191-196, ISBN 0-7803-6582-8

Monsef, Y. 1997. "*Modelling and Simulation of Complex Systems*", SCS International, San Diego, 1997, ISBN 1-56555-118-4

# MODEL VERIFICATION VALIDATION AND EVALUATION

# A Formal Definition of Simulation Validity

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

**KEYWORDS**
Validity, Abstraction, Experimental Frame, Modelling and Simulation

## ABSTRACT

The main objective of this paper is to propose a general approach for assessing the validity of Modelling and Simulation (M&S) used during the development of embedded systems. This approach is an effort to improve confidence in the use of a simulation whose results are often questioned without consistent justification. Considering that the validity of a simulation is never assessed in isolation but always in relation to a target user, we have defined the problem of validity as the applicability of a given M&S product to a given simulation objective of use.

## INTRODUCTION

In all fields, models are produced for the purpose of experimenting and predicting, attempting to approach and universalise the concepts of a system. Sociologists, biologists, ecologists and engineers all use the intellectual process of modelling with the aim of defining what constitutes and characterises a system and understanding its operation and its behaviour.

Simulation is already very widely used in engineering processes as an aid for decision-making. Estimating validity (correctness, fidelity, maturity, representativeness) is mandatory to formally evaluate the level of confidence that can be attributed to a simulation in view of its environment and its objectives of use. The question of the level of fidelity required for a need and, obviously, the effort required to reach this level of fidelity are the main issues for the embedded systems developments.

If we consider that a simulation is a model or a set of models subjected to an execution environment that gives life to this model over time, we create a distinction between the validity of the model and the correctness of the execution environment, i.e. the simulator. The execution environment adds further constraints, along with implementation errors, to the model validity. This environment is composed of complex computerised systems which are themselves composed of components (simulation models and real computers) and an infrastructure (real-time schedulers, digital processors, electronic interfaces and means of communication), all of which contribute to the level of simulation validity.

No level of confidence in a simulation has yet been formally defined and stated as an approved and standardised synthetic approach. This is obviously due to the recentness of the field. Numerous validation tests are carried out on all levels of simulation integration (non-regression testing, exhaustive testing of systems), with each engineering branch using its own resources and tools, but no coherent validation strategy has been defined.

In this contribution, we will be studying the M&S used for the development of embedded avionics systems. Within this context, a system is a physical element composed of two components: the avionics equipment itself (or "end system" in EIA-632 [1] jargon) and a set of systems used for the design, production, verification, operation, maintenance and recycling of the equipment ("enabling system").

We can associate with each of these systems a list of components that are physical objects, called the "system composition", as well as an environment. The same object cannot simultaneously belong to the system composition and the system environment. However, over time, it may move from the environment to the composition or vice versa. As regards an item of avionics equipment, the composition can always be given in terms of ports, processors, memories, specialised components, FPGAs, ASICs or analogue components.

These systems are characterised by a required level of reactivity and a relation to time (which may vary greatly depending on the field) which generally impose the capacity to memorise past behaviour in order to prepare present behaviour or even predict future evolutions.

Therefore, a system has structural properties related to static aspects (composition, connections, weight, dimensions, geometrical form) as well as behavioural properties related to dynamic aspects (processes occurring within the system, states, modes, actions on objects in the environment, reactions to actions of objects in the environment).

In relation to a Verification and Validation plan of a system, a simulation must be as close as possible to the system it represents while respecting the constraints of cost and timely availability. The simulations must be available before the systems themselves. If the level of validity is too low, the results required for the experiment can not be reached. If the level of validity is too high, time of modelling and calculation is unnecessarily spent.

Considering that the validity of a simulation is never assessed in isolation but always in relation to a target user, we have defined the problem of validity as the applicability of a given M&S product for a given simulation objective of use.

## DESCRIPTION OF THE APPROACH

We consider the problem of validity level as a hierarchy of model abstractions. We therefore proposed a model for describing properties of abstractions [2], called the conceptual model, which is the unifying language between user and developer of the simulation. It allows, first, to speak strictly the same thing when the term "validity" is mentioned, and secondly, to assess the compatibility between an expected validity level for the experience and a validity level provided by the simulation.

Then, we establish formal matching rules for mapping between an objective of use and a simulation described respectively as the conceptual model proposed. The problem of mapping is based on the principle that both are components, in the formal sense, i.e., they interact through their interfaces and only through their interfaces. We look at component-based engineering techniques to iteratively enrich the concept of "context validity" of a model by "symbolic concepts" and cover each property of the taxonomy of abstractions.

A conceptual model is prepared through abstractions. A variety of abstractions are used and they directly depend on the point of view from which the real world system is studied. The validity of a conceptual model is then evaluated from a given perspective. In DEVS and the M&S theory, B.P. Zeigler [3] proposed a way to supply a description of a model while questioning whether this model correctly reproduces the dynamic behaviour of the system from a given perspective.

The basic principle of the M&S process described by B.P. Zeigler involves the separation of the model and the simulator. The same principle is recommended by the MDA (Model Driven Architecture) approach proposed and supported by OMG [4]. The basic entities of the M&S process are the system, model and simulator (Figure 1).

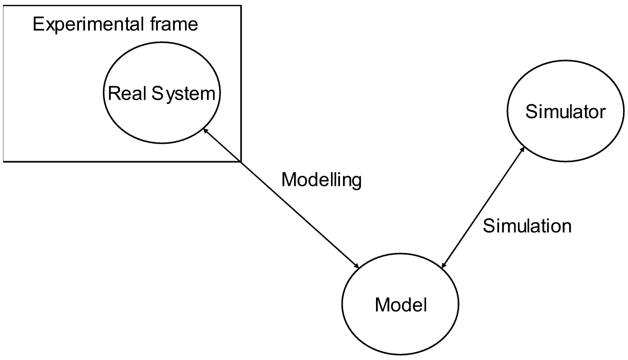


**Figure 1.** M&S process and its entities

The system is the real or virtual element used as a source of observable data and subject to modelling. The model, also called the system substitute, is a representation of the system. It is usually a set of instructions, controls, equations or constraints to generate its behaviour. The simulator is an IT system used to execute the model and generate its behaviour based on model instructions and injected inputs.

This set is reorganised so as to integrate the concept of an experimental frame. The experimental frame is a specification of the conditions in which a system is observed or experimented on. It can be seen as a system that interacts with the system of interest to obtain the data of interest in given conditions. It is an operational formulation of the objectives which motivates the development of the M&S application. There may therefore be several experimental frames for the same system and the same experimental frame may be applied to several systems. In fact, we can have several objectives or perspectives or have a single objective that motivates the modelling of different systems.

An experimental frame has three components: a generator which generates a set of input segments for the system; an acceptor which selects the data of interest of the system while monitoring whether the desired experimental conditions are complied with and a transducer which observes and analyses the output segments of the system.

It should be remembered that the experimental frame transforms the objectives, which are used to focus model development on a particular point of view, into specific experimentation conditions. A model must be valid for a system in such an experimental frame. An operational formulation of the objectives is produced by matching the observed variables (system inputs and outputs) with measurements of the system's effectiveness in accomplishing its function. These measurements are called the result measurements. Matching between observed variables and result measurements is carried out by the transducer.

The relationship between a model, a system and an experimental frame is modelling. The validity of a model is the fundamental concept of the modelling relation. Validity refers to the degree to which a model faithfully represents a system in an experimental frame of interest. The relation between a model and a simulator is simulation. The basic concept of this relation is the correctness of the simulator. A simulator correctly simulates a model if it guarantees to faithfully generate the model output values given the model state and the input values. This relation refers to the principle of separating preoccupations between model design and its implementation.

Besides the two basic relations presented above, B.P. Zeigler introduced two other relationships that are fundamental to our study: modelling as a valid simplification (valid abstraction in the terminology of F.K Frantz [5]) and the relation of model applicability to an experimental frame.

"Successful" modelling can be seen as a valid simplification. It is necessary to simplify or reduce the complexity of a model so that it can be run on a simulator, considered a limited computing resource. A preservation relation or system morphism establishes a correspondence between a "concrete" system and an "abstract" or simplified system, the abstract model being the substitute of the concrete model. B.P. Zeigler uses the concepts of base model and lumped model. In this case, the concrete model is a model with more capabilities, meaning that it can be used for a greater number of experimental frames. However, for a given experimental frame, the abstract model can be as capable as the concrete model. What must be remembered is that, for a given experimental frame, an abstract model must be as valid as the concrete model.

The applicability relation determines whether an experimental frame can be applied to a model. This relation is very important as it serves to state whether a use objective can be reached with a specific M&S application. Only models can implement experimentation conditions required by an experimental frame used to reach objectives and may possibly supply valid simulation results. The author also defines a derivability relation between experimental frames. This relation refers to the degree to which an experimental frame defines more restrictive conditions (which allow fewer observations) than another. Figure 2 below illustrates the morphism, applicability and derivability relations. Experimental frame EF4, not very restrictive, applies to models M1, M2 and M3. Model M4 is too abstract to accommodate EF4 as well as experimental frames EF1, EF2 and EF3 which are more restrictive than EF4. EF2 is applicable to Ml. EF3, which is less restrictive than EF2 is therefore also applicable. In this case, we can say that M1 is too concrete or complex for the given experimental frame but not less valid. No models can accommodate EF1.

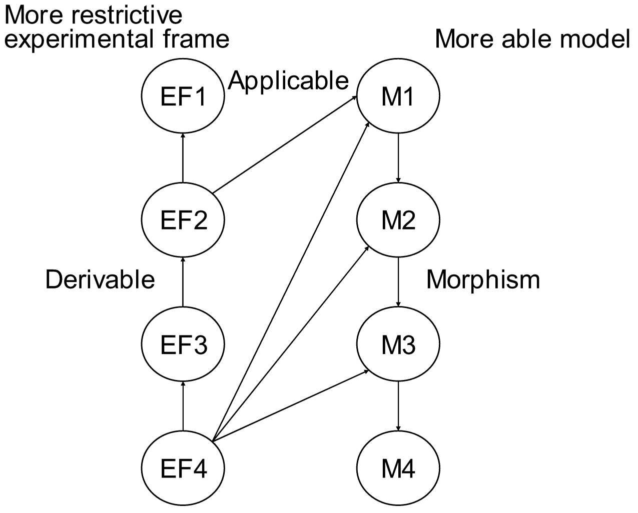


**Figure 2.** Morphism, applicability and derivability relations

## CHARACTERISING THE VALIDITY OF A SIMULATION

The aircraft is a system composed of a set of subsystems which may themselves be broken down into a set of subsystems. The aircraft development process is therefore broken down into a set of subsystem development processes. Higher-level requirements are broken down into lower-level requirements for each function or subsystem. The requirements are then translated until the end product is obtained. Then, moving back up through the aircraft description hierarchy, the end products, each one responsible for an aircraft function, are integrated to satisfy the requirements described on the highest level of the hierarchy.

The primary systems (ATA) of an aircraft (flight control laws, warnings, fuel, hydraulics, communication, navigation, engines, etc.) and their subsystems (sub-ATA) must exchange data to carry out their respective functions. A subsystem is thus designed to operate in a given environment or context, i.e. a reference experimental frame which, in our case, is the real aircraft.

A model is built through abstractions. As system development progresses, these models become increasingly concrete. We will therefore consider a set of models of the same system, prioritised by a morphism relation. In this hierarchy, a high-level model is a more abstract model than the low-level model, meaning that it can be used for a smaller experimental frame. The physical system is found on the lowest level of the hierarchy. It is composed of a set of software applications, distributed over a group of real computers, themselves distributed through a network and other physical elements. The most abstract level is the idea. Between these two levels, a multitude of models with different forms and different configurations are created: conceptual (requirements), formal (Matlab, Scade, Saber), executable (C, C++). These models are run on various platforms.

At a given moment, the development of a system requires the initiation of an M&S project. The use objective defines the system of interest, i.e. an isolated part of the system, modelled for a specific V&V plan. This use objective is then used to derive a specification for the models interfacing with the system of interest with a necessary and sufficient level of abstraction. The full, detailed definition of a model interfacing with the system of interest is not always the best solution. More abstract models can thus be used for suitable stimulations or observations of the system of interest.

As these interfacing systems are themselves undergoing development, a sufficiently detailed representation is not always available when required. Whether or not abstraction is desired, classes of abstractions must be identified to then be able to define the circumstances in which a model is a valid representation of the real system, i.e. the model usage domain. Describing such properties serves to ensure consistency and traceability between all the models of the same system and thus facilitate model updating.

As an M&S application is entirely dependent on a given objective of use and the validity of a model can only be measured for a given context, the evaluation must be conducted with the concept of experimental frame in mind. There are two experimental frames:

- SOU: Simulation Objectives of Use (SOU), the experimental frame that describes the way in which experimentation of the system of interest will be performed. This specification will be paired with the executable model to generate the expected results. This experimental frame must be enriched to specify the acceptable abstractions.
- SDU: Simulation Domain of Use (SDU), the experimental frame that describes the usage domain of a model, i.e. its properties and its limitations.

Through a verification and matching process between these two experimental frames, we propose verifying the compatibility between the scenario and the model. A simulation is valid if an SOU is applicable to an SDU. In this case, the concept of applicability is used with reference to B.P. Zeigler's concept mentioned previously.

This approach is used to evaluate validity:

- A priori: defining the necessary and sufficient SDU to satisfy a given SOU.
- A posteriori: defining whether an SDU already developed satisfies a given SOU

## FORMAL DEFINITION OF SOU/SDU APPLICABILITY

The principle of component-based software engineering (CBSE) consists in composing a software system which meets given requirements by using components. A software component is [6] a composition unit with specific contractual interfaces and explicit contextual dependencies. A software component can be deployed autonomously then subjected to a composition by third parties. Through its interfaces, the component is connected to its environment. The objective of the formal approaches of component-based software engineering is to specify the properties of a component for its interfaces. This consists in enhancing the interfaces by supplying a signature and a specification of the component [7]. The signature describes information on the type of component. It has the same role as a type signature in programming language. For example, it defines the type of input/output of a function or method. This includes the name of the function and the number of parameters and possibly the type of data returned by the function. The signature covers the static aspects of the component. The specification describes the dynamic behaviour of the component. More particularly, it characterises its semantics more than a simple signature, e.g. the order in which the component functions must be called up or the communication protocol used. A component can be specified using several methods. A common method of describing the specification of a component is to use Boolean assertions called pre- and post-conditions for each service provided, together with invariants to define general properties for consistent use of a component. A specification is then interpreted as follows: a client may draw up a request to a supplier only if the service required is in a state that does not contravene the invariant and the precondition related to the service. In return, the supplier guarantees that what is specified in the post-condition will be complied with, along with the invariant. The rights and duties of the client and supplier are clearly identified. This method initially originated from the Eiffel language [8], under the name of design by contract [9] [10]. It is now used in a large number of languages such as Java and also in UML, notably via OCL [11]. In [12], the authors use interface automata [13] to describe the specification of a component.

Specification matching is then used to define whether two software components are syntactically and semantically linked.

This serves to resolve a set of miscellaneous questions [7]:

- *Recovery*: How to recover a software library component based on its semantics rather than its syntactical structure?
- *Reuse*: How to adapt a software library component to suit the needs of a given subsystem?
- *Substitution*: When should a component be replaced by another one without affecting the observable behaviour of a system?
- *Subtype*: When is an object a sub-type of another object?

Given a set of experimental frames EF and A, a set of attribute values (e.g. bool, string, etc.). A property P is a function P: EF —> $2^A$, here $2^A$ is the set of all subsets of A. A property therefore assigns a specific attribute to an experimental frame. For example, a property is a function P which assigns to an experimental frame ef the names of input variables, e.g. P (ef) = altitude, MachNumber.

Given SOU and SDU, two components whose signature and specification are characterised by P(SOU) and P(SDU), respectively, the problem of applicability of a SOU to a SDU is defined by:

Applicability: P (SOU), C, P (SDU) —> {0.1}

Given a property for an experimental frame SOU which characterises a set of simulation scenarios and the acceptable abstractions for reaching a use objective, a property for an experimental frame SDU which characterises the usage domain of a simulation product, and $c \in C$, an acceptability criterion, Applicability returns the value "1" if the SOU is applicable to SDU for this criterion, else "0".

## ILLUSTRATION

We propose an illustration of this approach on the property called scope [2]. D.S. Weld [14] defines the scope of a model as the range of the system the model describes. A model has a larger scope than another if it gives a broader description of the system. Changing the scope of a model is equivalent to redefining the boundaries between the system (described by the model) and its environment. So, selecting the scope of the model implicitly includes selecting the exogenous parameters and determining their values.

An exogenous parameter is an external parameter on which the system depends. An endogenous parameter is an internal system parameter. To illustrate this property of scope, we will consider system A (Figure 3a) which has a set of endogenous parameters, represented by the "x"s and a set of exogenous parameters, represented by the circles. By reducing the scope of the model of system A, we get system B (Figure 3b) with a new (smaller) set of endogenous parameters and a new (larger) set of exogenous parameters.

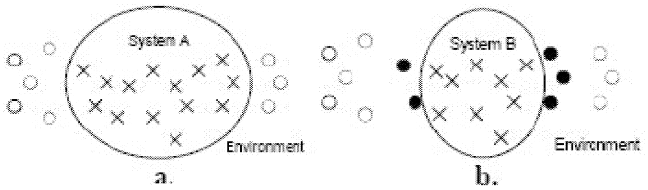


**Figure 3.** Models with different scopes

The new exogenous parameters (black circles) then become parameters that can be controlled and/or observed by the simulation user to properly carry out his experimentation. They now belong to the experimental frame. We thus define the scope of an M&S application by the set of exogenous parameters of the system of interest.

The applicability criterion related to *scope* property consists in making sure that the scope of the M&S product covers the scope required by the simulation use objective.

At the lowest level of the specification hierarchy [2], an experimental frame is defined by the structure

$$EF =< T, X_{EF}, Y_{EF} > \text{ where}$$

T, is the time base
$X_{EF}$, is the set of input variables
$Y_{EF}$, is the set of output variables

We define the scope of an experimental frame EF as the set of input variables $X_{EF}$ which influence the system and the set of output variables $Y_{EF}$ which are controlled by the system:

$$scope_{EF} = X_{EF} \cup Y_{EF}$$

SOU is applicable to SDU according to their respective scopes if, and only if

$$scope_{SOU} \subseteq scope_{SDU}$$

If we consider the experimental frame of an experimentation defined by SOU= (T, $X_{SOU}$, $Y_{SOU}$) where

$T = c \bullet N$ for example (discrete-time)
$X_{SOU} = \{X_{SOU1}, X_{SOU2}\}$ the set of variables observed
$Y_{SOU} = \{Y_{SOU1}, Y_{SOU2}\}$ the set of stimuli injected

Consider a simulation model whose usage domain is defined by SDU = (T, $X_{SDU}$, $Y_{SDU}$) where

$T = c \bullet N$ for example
$X_{SDU} = \{ X_{SDU1}, X_{SDU2}\}$ the set of input variables
$Y_{SDU} = \{Y_{SDU1}, Y_{SDU2}, Y_{SDU3}\}$ the set of output variables

The corresponding simulation product is presented in Figure 4a. Figures b., c. and d. illustrate the other possible configurations and their consequence on applicability.

In the case of Figure d., we consider that the scope of SOU is applicable to the scope of SDU. However, it is necessary to make sure that there is no dependence between $X_{SDU3}$ and $Y_{SDU1}$ or $Y_{SDU2}$ , in which case the simulation results could be biased.

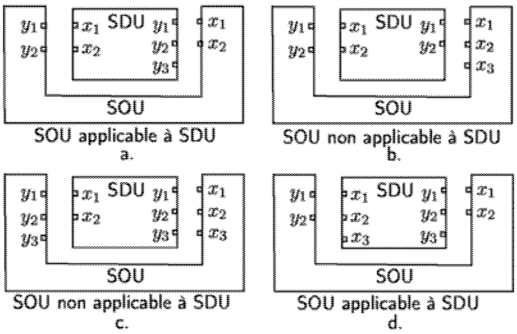

$y_1$ $y_2$ $x_1$ $x_2$ SDU $y_1$ $y_2$ $y_3$ $x_1$ $x_2$
SOU
SOU applicable à SDU
a.
$y_1$ $y_2$ $x_1$ $x_2$ SDU $y_1$ $y_2$ $x_1$ $x_2$ $x_3$
SOU
SOU non applicable à SDU
b.
$y_1$ $y_2$ $y_3$ $x_1$ $x_2$ SDU $y_1$ $y_2$ $y_3$ $x_1$ $x_2$ $x_3$
SOU
SOU non applicable à SDU
c.
$y_1$ $y_2$ $x_1$ $x_2$ $x_3$ SDU $y_1$ $y_2$ $y_3$ $x_1$ $x_2$
SOU
SOU applicable à SDU
d.

**Figure 4.** Scope applicability of a simulation

We will illustrate the characterisation of this property with an example whose global system is a vehicle whose speed is adapted according to the distance separating it from the vehicle preceding it. We have decided to study the vehicle cruise controller in more depth [15]. In these conditions, we must define the exogenous parameters of the cruise controller. All other components of the global system now belong to the experimental frame to represent the environment of the cruise controller. Some of these elements can be abstracted or simplified if their contributions are not relevant to the cruise controller. The experimental frame then consists of input stimuli for the speed of the vehicle of interest (speedV1), the distance (distance) and the speed of the previous vehicle (speedV2). The simulation user makes the assumption that the aerodynamic form of the vehicles is not essential. The user wishes to observe the position of the throttle. Figure 5 shows the boundary between the experimental frame and the model of the system of interest for a cruise controller M&S application.

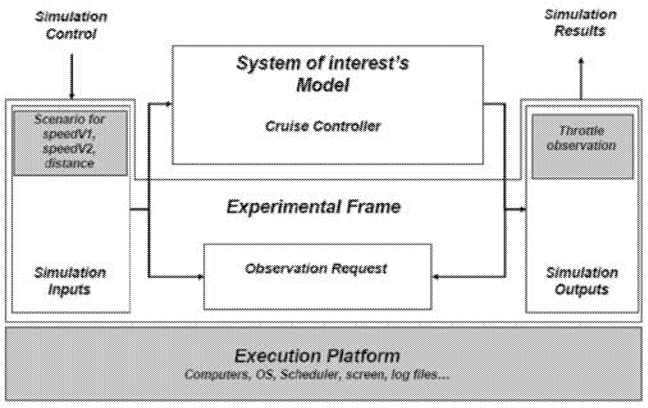


**Figure 5.** Relations between the model of the system of interest and the experimental frame for validation of a cruise controller

Now imagine that it is no longer the cruise controller being studied but its integration in the vehicle. In this case, speedV1 becomes an endogenous parameter of the system of interest. We can then imagine that we no longer inject the speed of the vehicle in the simulation but rather other (logical or physical) control elements such as the gas and brake pedals. New boundaries between the experimental frame and the model of the system of interest are defined (figure 6).

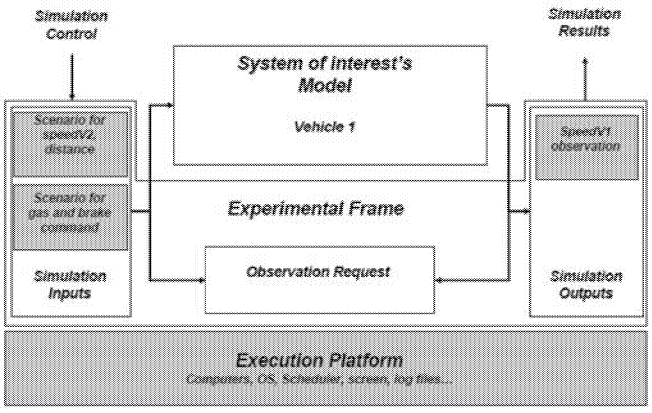


**Figure 6.** Relations between the model of the system of interest and the experimental frame for validation of a vehicle

## CONCLUSION

The aim of this study was to propose a general approach for evaluating the validity of an M&S application used for the development of embedded systems. This approach is part of a procedure intended to improve confidence in the use of a simulation, whose results are often questioned with no coherent justification. One contribution of this study was to define a context for justification.

By considering that the validity of a simulation is never evaluated directly but always with respect to an objectives of use, we have defined this issue of validity as the applicability of an objective to a simulation model.
The solution proposed was a validity evaluation system based on:

- an evaluation-oriented model describing the properties of the embedded systems and the simulation, i.e. used to match model elements.
- formal matching rules between a use objective and the usage domain of a model.

We based this model on the concept of experimental frame. We used this concept, proposed in the M&S theory, to address the problem within a well-founded methodological framework. It is important to note that the experimental frame can also describe, in addition to an experimentation framework and a validity framework, a reuse framework (description of the new use or experimentation context) and an interoperability framework (description of the way the model will communicate).

We have illustrated our approach on scope property. Further works aims at using the same approach, i.e. enriching the concept of experimental frame, for each property describing the level of abstraction of a model that we have informally developed in [2], including dynamic properties.

## REFERENCES

[1] Electronics Industries Alliance, *Processes for Engineering a System : EIA-632*, Janvier 1999.

[2] Vincent Albert, Alexandre Nketsa, Mario Paludetto, Marc Courvoisier, *Criteria and methods to establish the valid interaction of a simulation and its intended purpose*, European Simulation and Modeling Conference, pages 29-36, 2007.

[3] Bernard P. Zeigler, Herbert Praehofer & Tag G. Kim. *Theory of modelling and simulation*. Academic Press, San Diego, California, USA, 2000.

[4] Object Management Group, *Model Driven Architecture*, 2001. Available at http://www.omg.org/mda/

[5] Frederick K. Frantz, *A taxonomy of model abstraction techniques,* 27th conference on winter simulation, pages 1413-1420, 1995.

[6] Clemens Szyperski, *Component software: beyond object-oriented programming,* Addison-Wesley Longman Publishing Co., 2002.

[7] Amy Moormann Zaremski and Jeannette M. Wing, *Specification matching of software component,* ACM Trans. Eng. Methodol., vol. 6, no. 4, pages 333-369, 1997.

[8] Bertrand Meyer, *Eiffel: a language and environment for software engineering,* Journal Syst. Soft., vol. 8, no. 3, pages 199-246, 1998.

[9] Antoine Beugnard, Jean-Marc Jézéquel, Noël Plouzeau & Damien Watkins, *Making Components Contract Aware*, Computer, vol. 32, pages 38–45, 1999.

[10] Cyril Carrez, Alessandro Fantechi & Elie Najm. Behavioural, *Contracts for a Sound Assembly of Components*, FORTE, pages 36–39, 2003.

[11] Object Management Group, *UML 2.0 OCL Specification*, 2004. Available at http ://www.omg.org/docs/ptc/03-10-14.pdf

[12] Edward.A. Lee & Ye Xiong, *Behavioural Types for Component-Based Design,* Technical Memorandum UCB/ERL M02/29, 2002.

[13] Luca Alfaro & Thomas A. Henzinger, *Interface automata*, Ninth Annual Symposium on Foundations of Software Engineering (FSE), ACM, pages 109–120, 2001.

[14] Daniel S. Weld, *Reasoning about model accuracy*, Artif. Intell., vol. 56, no. 2-3, pages 255–300, 1992

[15] S. Schulz, J. W. Rozenblit & K. Buchenrieder, *Towards an Application of Model-Based Codesign : An Autonomous, Intelligent Cruise Controller*, IEEE Conference and Workshop on Engineering of Computer Based Systems, pages 73–80, 1997.

# CONCEPTS FOR MODEL VERIFICATION AND VALIDATION DURING SIMULATION RUNTIME

Wilhelm Dangelmaier
Robin Delius
Christoph Laroque

Business Computing, esp. CIM

Heinz Nixdorf Institute
University of Paderborn
Fuerstenallee 11, 33102 Paderborn, Germany

Matthias Fischer

Algorithms and Complexity

Heinz Nixdorf Institute
University of Paderborn
Fuerstenallee 11, 33102 Paderborn, Germany

**KEYWORDS**
Dynamic Modelling, Validation, Verification

## ABSTRACT

Modern companies are nowadays confronted with an increasing demand of multiple products, where they need to perform more flexible every day. Cost-intensive decisions are to be confirmed in short times, in order to minimize risks and secure efficient production programs as well as material flows. Tools for this digital planning via simulation methods are one well established possibility to receive decision support. Nevertheless, the creation of the necessary simulation models is a complicated and error-prone process, where complexity of modeling, validation and verification depends on the used tool and its functionalities. This paper presents implemented concepts for an innovative user support in his tasks of verification and validation of simulation models during the execution of a simulation run. Time-intensive procedures like stopping simulation, parameterization and restarting within the problem analysis are simplified. So the user is able to focus on the real problem solving task.

## 1 INTRODUCTION

In times of proceeding industrialization and commercialization modern companies are facing markets pressure more than ever. For short as well as long-term success the needs for high flexibility to react on market changes and the customer demands is constantly growing. In line with the requirements to produce highly qualitative goods in time, each single processes within the affected company needs to be synchronized perfectly. By securing cost-intensive decisions via decision support methodologies like simulation, these results have to be available as fast as possible. Nevertheless, the necessary creation of the underlying models is a complex, time-intensive and error-prone process. With focus on the exemplary approach of model creation, described e.g. in (Sargent 2007), there is plenty of work to do until the finished, correct and validated model is available. Especially the late occurrence of errors during simulation execution is a problem, because late corrections need a “stepping back” to earlier steps within the modeling process. This paper introduces an approach to close this gap. Since actual solutions are separating modeling in different process steps, the presented approach embraces the earlier steps of modeling into the simulation execution. This results in a faster problem solving solution with fewer complications. Consequently the concept also focus on the continuation of an executed simulation run in order to achieve valid simulation results even after changing the underlying model.

## 2 STATE OF THE ART

Modeling as a process and its challenging nature is sometimes by far underestimated .It consists of the analysis of the underlying system, where all relevant parameters and processes have to be identified and gathered. Followed by the process mapping, which transforms the previous insights into a formal model relation, a validation and verification needs to be performed before a final model can be used for simulation experimentation. Basically a model

- is the mapping of a real system or environment,
- is not necessarily a complete representation of the reality.

The last statement is important because modeling is done by a simulation expert, who can choose the level of abstraction used for the model based on his experience. Nevertheless, a model is thereby also an interpretation of the modeler and of his perspective. So, a model can be correct for an intended purpose, but it does not have to match a model created by another expert for the same purpose. Depending on the interpretation of the modeler, many different models can exist for the same purpose. Just with focus on the process of modeling, there exist classically three views on realities’ examination. By choosing one of these views for the analysis process, this decision has a huge impact on the resulting model. These views describe an abstract worldview respectively an abstraction of the properly perception of the system by the observer (see (CarsonII 2005), (Overstreet and Nance 2004), (Pidd 2004)). These views are the Event-Scheduling-View (event view), Activity-Scanning-View (activity view) and Process-Interaction-View (process view).

## 3 MOTIVATION

With the intention to combine the detached steps of modeling, including validation and verification, and simulation, this work describes the developed concepts in order to support the simulation expert in his task of building correct models, especially regarding those modeling task, which might lead to a significant state change in a simulation run. The tasks composed under the term of modeling are

- the creation of new simulation elements
- the instantiation of simulation elements
- the embedding of instantiated simulation elements
- the deletion of embedded or instantiated simulation elements
- the modification / expansion of created, embedded or instantiated simulation elements
- the replacement of created, embedded or instantiated simulation elements

In sum, all of these tasks have multiple impacts on a concrete simulation state. They need to be considered closely, when trying to preserve a valid simulation state during an interactive modeling approach. Looking at a simulation state, at any point in time only one concrete state occurrence exists. This concrete occurrence is only a subset of the larger state space, which includes all possible states. Problems occur, when the existing state space is reduced or partly replaced. Former existing states might not be reachable any more, which can cause an incomplete simulation run. In addition, executing an incorrect state transition can cause an incorrect state space, which could still allow to finish a running simulation, but with incomplete or erroneous results.

In traditional simulation and modeling, the modification of the functional behavior has to be done offline. The simulation needs to be stopped and restarted, just to see the results of the modifications. Instead of that, by using the approved concept, the user can make the modifications online and continue the simulation with the benefit of seeing his changes immediately and the opportunity of reaching a valid simulation result faster, more reliable and therefore in sum less costly.

## 4 CONCEPTS FOR METHODS IN MODELING

This section describes some ideas in the area of simulation modeling and how simulation tools and methodology could be enriched for an interactive validation and verification of simulation models during their execution in a simulation run.

### 4.1 Modeling supporting methods

Basic idea of some of the described approaches is the fact, that today it is more or less a standard for simulation tools, that the simulation expert can develop building blocks in a high level language like C++ or Java. In addition users can, without knowledge about complex programming languages, design highly functional building blocks by arranging simple predefined functions. This can be achieved, since commercial tools offer typically their own simulation language, which can be learned by an inexperienced user in a short period of time. These tools also include a large variety of predefined building blocks, which cover most of the standard applications in simulation. A great advantage is that these building blocks are already validated. On the other hand, by using predefined functions or building blocks the question occurs, if these elements are fully suitable for the special purpose they are intended for.

A possible solution is a typification of predefined elements. The user is free to adjust these predefined elements in a restricted way, which will not violate their type. The modified building block would be an extension of the old one, with extended functionality.

**Automatic Functional Analysis**

With focus on the internal behavior of each simulation element there exist multiple problems to be dealt with. The presented example of typified elements is an ideal vision which seems to be achievable. By typifying a predefined element for example as a conveyor, the user must be assured, that its behavior is correct. If it behaves like a drill, a wrong typification was made, which will end in an incorrect model. Here the cognitive output of the modeler comes into play. Comparing the design for a specific element, done by two simulation experts, both elements can behave the same, but are implemented in completely different ways. A granular description of the detailed functions would be benefiting, due to the fact that this is the base for an automatic functional analysis. Staying with the example of the conveyor-element, its behavior could be splitted into single steps, whereas their composition would describe it completely. For example, these steps could be ‘workpiece entering’, ‘switching on motor’, ‘workpiece leaving’ and ‘switching off motor’. ‘Workpiece entering’ describes the state, where the workpiece enters the area of operations, from which point on its ment to be moved by the conveyor. The next step would be, that the conveyors’ motor is switched on, so that its band or belt starts moving, described by ‘switching on motor’. Once the workpiece leaves the conveyor, this state is described by ‘workpiece leaving’. The last step in this simple example is ‘switching off motor’ when the conveyors job is done. Definitely this description of a conveyors’ functional behavior is not nearly complete, it even does not handle any uncertainties, but it is good enough to point out the general idea. These steps can be described in such a granular manner, that an automatic analysis of the functional behavior comes into approach. Since an element of a simulation environment is still part of a computer program, without additional restrictions, the question, if a program will halt within finite time, cannot be answered. According to that, there is no way to derive from an elements programming code, how it will behave functionally. Focusing on programming languages with reduced functional complexity some automatic analysis of the functional behavior becomes possible. A complete decoding of the scope of operation of an element still is not possible, but statements like ‘the element is a token-creating building block’ or ‘the element alters tokens’ are conceivable. So the size of restrictions defines the area of applicable automatisms. Some tools (see Krahl, D. 2003) distinct subsets of its predefined elements as ‘object moving’ or ‘object changing’. But to achieve such a functional description of an element, at first a formal definition of the building block is necessary. Such a description must include the components of a building block as well as its functional structures.

In the last years some research work concerning this topic (see (Roehl and Morgenstern 2007) or (Gustavson and Chase 2007)) was done. Nevertheless, there is still open work to achieve a standardized definition. Complete modeling freedom stays in contrast to a highly error-prone modeling process. With focus on other objectives like reusability and simplicity, the need for a more restrictive model development seems to be more benefiting. Restriction could mean that the developed models need to be minimal in their functional components, like atomic and indivisible methods. Such a component on its own has no great power, but the combination of these components becomes powerful. By applying these minimal components an automatic functional analysis becomes possible. Only the minimal components have to be analyzed and the whole model could be meta-tagged by the evaluated functional behaviors of its consisting elements.

## Analysis, Verification and Validation

Before establishing a sustainable statement about the correctness or quality of a simulation model, some aspects of the internal logic and structure must be observed. When designing a building block conceptually, its error-proneness can be reduced by using minimal design-pattern. Resulting from this minimum approach, it is easier to evaluate incorrect components. Nevertheless, there is still the need for additional methods, which will help in error prevention. Sargent describes (see Sargent 2007) how methods for validation and verification should look like ideally.

There has to be a clear breakup between analysis and validation, since at first glance both seem to be very close. In this context, the way to the objective of analysis should extend the validation and be used as a supportive method. Validation means checking the model and its components for correctness, which includes also structural and logical correctness. Structural correctness describes the concrete implementation of code, whereas logical correctness describes the allegiant implementation of the correct behavior, so that produced results are also deductable in reality. Only if both points are checked and no errors were detected, the building block or entire simulation model may be accepted as correct. Analysis in its original intention means to face a problem of concern, under this context maybe the comparison of different functional behaviors for an element. For a validation of elements or entire models, the user can choose the classical alternative and test any single element under special criteria, by generating input, which has to be processed by the element and for which the resulting output is already known previously.

Here, an automatism helps to reduce the necessary time significantly. For all elements of the model, configurable testing sources and sinks can be generated, which must be connected to the elements. For each element, a testing source is connected to its input ports and a testing sink is connected to its output ports. The source is sending predefined input to the element in predefined time intervals, so that a correct surrounding is reproduced. When such a testing scenario is finished, the attained results can be evaluated with regard to a correct building block behavior. With this technique, a model can be checked for its logical correctness step by step with less expenditure of time. In addition, the reusability of elements is increased. This process of validation establishes the opportunity to gain supplementary results concerning the performance of an element or a model. Especially in the case of checking alternative strategies, methods of supportive analysis seem to be very helpful, due to fact that their results can help drawing conclusions from the source of failure.

## Simulation Breakpoints

An additional method is the use of breakpoints. It allows the user the definition of simulation states, where the simulation has to pause automatically. In combination with the so called simulation-debugger, the user can continue the simulation step-by-step and observe relevant simulation or model parameters. In contrast to the area of software engineering, where concrete code sections are associated with breakpoints, the firing of events seems to be a supposable counterpart when applying them to simulation. This seems to be obvious due to the fact, that they control the simulation execution with close relation to the simulation state.

## Object Flow and Visualization Layer

When searching for errors, the knowledge about the movement of objects can be very helpful. For example, a common error is the slack flow of objects because of missing or wrong placed connections between the simulation elements. By logging the object flow, a mapping can be created, which shows the objects' paths as well as their utilization. Such a logging function can be extended to a monitoring, which gives the user a greater insight into the dependencies by observing defined operation figures. For the field of model validation this information might help to find erroneous sections in the model itself. In the field of analysis, a simple visualization of the object flow can give interesting information, for example on different lengths of existing paths or their concrete capacity depending on the online data (e.g. Max-Flow algorithms). This might help to find bottlenecks, which are often the reason for an incorrect designed or parameterized. In combination with the modelers' knowledge about the model and its intended functionality, it becomes easy to find incorrect sections of a model.

## Event-Graph

Using the event view in simulation model development, the modeler may create a simulation event graph (see (Schruben 1983),(Schruben and Yucesan 1988), (Schruben and Yucesan 1989), (Schruben 1992), (Schruben and Yucesan 1992), (Schruben and Yucesan 1994)), which represents the event dependencies. This graph can serve as a source when creating and programming the building blocks and their events. This describes a classical and not often practical approach when creating a simulation model. Typically, an event graph does not exist. This is not necessarily a problem, since the dependencies of the elements and events can also be pointed out by other means. But without such an event graph, the modeler misses an important and very helpful tool for model validation and analysis. With the intention to handle all possible states of the different elements and their relations among each other, a large complexity arises. It is evident, what dimensions an event graph can obtain, when more complex models are designed.

Additional to the obvious advantages of an event graph as a supportive method for the simulation expert, it can be also very helpful during model simulation. Here, the method of event logging can be used in combination to build an 'online event graph', which can be used for a matching with the 'offline event graph'. This identifies non-firing events, which are intended to be executed during simulation runtime.

### 4.2 State Preserving Functions

With focus on a complete convertibility of simulation models during runtime, mechanisms are necessary for preserving the simulation state. These mechanisms can be either automatic or manual, depending on the problems the user is facing. A fundamental problem during model modification is the transformation of the old simulation state into a new one, since it is preferable to have not only a valid but also a correct state. Here valid means that the new simulation state is located in the existing state space, whereas correct means, that a concrete mapping exists, which leads to a correct simulation result. Often a valid state may be sufficient.

A first motivation for modification is the validation itself, which includes locating incorrect parts of the model and leads to a correct model by the use of model analysis. Additionally, the comparison of alternatives, which has the overall goal of identifying better system solutions, may be solved. With focus on these two motivations, a look on the consequences for the simulation state and its transformations seems appropriate.

Assuming, that the goal is to find and correct errors in the model, this implies that the model did not perform in its intended way. This means, that it did not reflect the intended system correctly, which directly leads to the assumption, that its produced results would not be applicable. In this respect, a modification during runtime helps producing a 'more correct' model behavior while already simulated periods have generated erroneous results. Then, only a complete restart of the simulation can generate correct overall results. But why should it be intended to create a correct state, which will hopefully lead to a correct result, when it is known, that a complete correct simulation result cannot be achieved because of already simulated wrong model sections? Here, a valid simulation state, based on a correct state transformation seems to be a more preferable goal, since this will lead to a modified model, which behaves correct from that point in time, when the modification has been done.

An elementary and necessary functionality is a method for saving and exchanging states. Ignoring the fact, that saving is memory and time consuming, an implementation is no real problem. But often a modification affects more than just a single component. Especially when affecting elements like tokens, which are representing workpieces or materials, it is very hard to apply automatic methods, because in these cases knowledge about the whole model and its intention is necessary. What are the consequences for tokens inside a building block when it is modified or exchanged? What does the modification of a building block imply for tokens inside of the simulation? These are some questions to deal with.

Possible solutions can be incremental state saving (see (Steinman 1993), (Feng and Lee 2006)) or an event-anti-event-based method. The incremental state saving works with predefined time intervals, at which boundaries a complete simulation state is saved. When necessary it allows loading an earlier simulation state, based on the states saved before. The idea is, that modifying a component initiates an automatic state rollback where the earliest state has to be chosen, before the component first interacted with the rest of the simulation model, which means for example the first time when a token passed this component. After modification, the simulation starts with the exchanged state, because now, no token has passed the component.

A method similar to the incremental state saving is the event-anti-event based approach. Here, no state saving is realized like before. Instead, the modeler has to differentiate between events, which allow a forward progress of the simulation and events which manage a backward progress. When simulating a model, every execution of a forward event is logged which allows to execute the backward events in reverse order. The advantage of this method is, that the system must not provide mechanisms for saving and transforming a state, because the simulation expert is in charge to provide the correct event pairs. This describes also the soft spot of the method, because more events exist and errors can creep in and cause incorrect functionality. Likewise the process of modeling is stretched in time because the modeler has to define twice the size of events as before.

## 5 PROOF OF CONCEPT

The developed concepts have been applied in research of the event-oriented simulation environment d³FACTinsight at the University of Paderborn. Within the research work, the simulation kernel has been enhanced for achieving a full interchangeability of the simulation elements during simulation runtime. Based on that, a complete 3D-modeling and visualization tool was created, which encapsulated parts of the described functions and concepts. Figure 1 shows the main view of the visualization and editing component. A list of all stationary components gives the opportunity to navigate quickly to the components position.

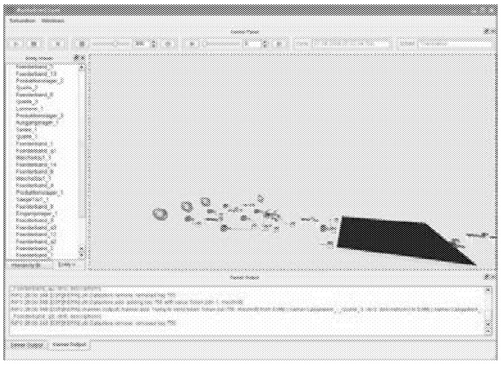
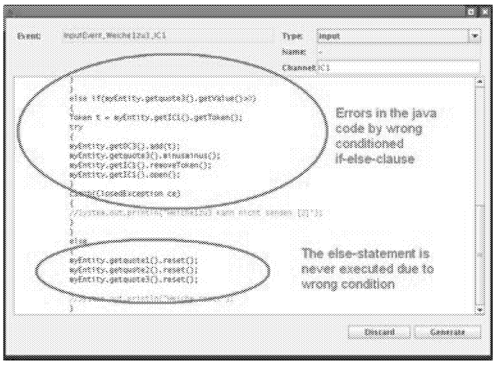


**Figure 1: 3D-Visualization and Editor**

The model itself is displayed in the center. The user can move in all directions by using his keyboard and edit all components inside his perspective. The tool has been used for the creation of different models and helped to resolve a lot of common modeling errors in less time. Regardless if these errors are results of incorrect modeling or a correct mapping of the reality, applying the introduced methods leads to the problems' origin and can be used for model correction.



Event: InputEvent_Weiche12u3_IC1

Type: input

Name: -

Channel: IC1

```
}
}
else if(myEntity.getquote3().getValue()>0)
{
Token t = myEntity.getIC1().getToken();
try
{
myEntity.getOC3().add(t);
myEntity.getquote3().minusminus();
myEntity.getIC1().removeToken();
myEntity.getIC1().open();
}
catch(ClosedException ce)
{
//System.out.println("Weiche12u3 kann nicht senden [C]");
}
}
else
{
myEntity.getquote1().reset();
myEntity.getquote2().reset();
myEntity.getquote3().reset();
}
```

Errors in the java code by wrong conditioned if-else-clause
The else-statement is never executed due to wrong condition
Discard
Generate

**Figure 2: Editing java code while runtime**

For example checking the code (see Figure 2) may show wrong statements in the programmed events and the user can modify the java code directly, which causes the system to load the new code into the system. Immediately the components incorrect functionality is overwritten by its modified one and the user can observe that the objects are moving again without restarting the simulation.

## 6 CONCLUSION

The insight, that modeling is a complex and error-prone process, was the origin for this work. The task, to identify only the relevant and to ignore redundant information seems to be trivial but holds an enormous complexity on closer examination. Likewise is the creation of the simulation program, for which the model serves as a template, a creative process which builds on the knowledge and the experience of the involved experts. Existing methods are for offline-use, which can help to build and evolve the model, but still are time-consuming. Here, the presented approach tries to close the gap and develop concepts with focus on validation and verification during simulation runtime.

The introduced methods serve as a conceptual draft for possible modifications during runtime with focus on extensive modifications but without interrupting the simulation. The implementation of the methods and integrating these into the d³Fact insight-simulator has been a first step. In further research, we want to delve into this topic by exploring more concepts and elaborate the introduced ones. Also is the further development of the visualization- and modeling-tool part of future work, since this is the interface, which gives the user access to these methods.

## REFERENCES

CarsonII, J. S. 2005. Introduction to modeling and sim-ulation. Proceedings of the 2007 Winter Simulation Conference.

Feng, T. H., and E. A. Lee. 2006. Incremental checkpointing with application to distributed discrete event simulation. Proceedings of the 2006 Winter Simulation Conference.

Gustavson, P., and T. Chase. 2007. Building composable bridges between the conceptual space and the implemen-tation space. Proceedings of the 2007 Winter Simulation Conference.

Krahl, D. 2003. Extend: An interactive simulation tool. Proceedings of the 2003 Winter Simulation Conference.

Krahl, D. 2007. Extendsim7. Proceedings of the 2007 Winter Simulation Conference.

Nordgren, W. B. 2003. Flexsim simulation environment. Proceedings of the 2003 Winter Simulation Conference.

Overstreet, C. M., and R. E. Nance. 2004. Characterizations and relationships of world views. Proceedings of the 2004 Winter Simulation Conference.

Pidd, M. 2004. Simulation worldviews -so what? Proceed-ings of the 2004 Winter Simulation Conference.

Roehl, M., and S. Morgenstern. 2007. Composing simulation models using interface definitions based on web service descriptions. Proceedings of the 2007 Winter Simulation Conference.

Sargent, R. G. 2007. Verification and validation of simulation models. Proceedings of the 2007 Winter Simulation Conference.

Schruben, L. 1983. Simulation modeling with event graphs. Communications of the ACM.

Schruben, L. 1992. Graphical model structures for discrete event simulation. Proceedings of the 1992 Winter Sim-ulation Conference.

Schruben, L., and E. Yucesan. 1988. Simulation graphs. Proceedings of the 1988 Winter Simulation Conference.

Schruben, L., and E. Yucesan. 1989. Simulation graph duality: A world view transformation for simple queueing models. Proceedings of the 1989 Winter Simulation Conference.

Schruben, L., and E. Yucesan. 1992. Structural and behav-ioral equivalence of simulation models. ACM Transac-tions on Modeling & Computer Simulation.

Schruben, L., and E. Yucesan. 1994. Transforming petri-nets into event graph models. Proceedings of the 1994 Winter Simulation Conference.

Steinman, J. S. 1993. Incremental state saving in speedes using c++. Proceedings of the 1993 Winter Simulation Conference

# SRN Model for Performance Evaluation of TCP Sessions Sharing Bottleneck Links in WAN

Osama S. Younes
School of Computing Science
Newcastle University, UK
*osama.younes@ncl.ac.uk*

Wail S. Elkilani
Faculty of Computers and Information
Menoufia University, Egypt
*welkilani@gawab.com*

Nigel Thomas
School of Computing Science
Newcastle University, UK
*nigel.thomas@ncl.ac.uk*

## ABSTRACT

Understanding and evaluating TCP behaviour remain a challenging problem, because of the complexity of the protocol itself and the inherent complexity of the interactions between the protocol and the network. Several analytical models of the TCP behaviour have been proposed recently. Most of these models assume a fixed number of persistent TCP connections. A few of the recent studies on TCP look at non-persistent TCP connections but with many limiting assumptions. Moreover, nearly all non-persistent studies are macroscopic and not scalable. This paper introduces a scalable stochastic reward Petri nets (SRNs) model for non-persistent TCP sessions that share two bottleneck links in a wide area network (WAN). The suggested model is constructed in a microscopic approach which captures the essential protocol features of the congestion control mechanism used by the TCP Reno protocol. Several limiting assumptions have been relaxed by this model. The interference from other non-persistent sources is introduced to the model in an efficient way. Several performance metrics have been captured and compared to simulation results.

**KEYWORDS**
TCP, Petri Nets, congestion control, modelling, TCP performance.

## INTRODUCTION

With the growing size and popularity of the Internet as a medium for exchanging information and conducting business, there has been growing interest in modelling and understanding Internet traffic. Accurate modelling of Internet traffic is also important from the perspective of deploying differentiated services since it is likely that best-effort traffic will comprise a significant portion of the internet traffic in the foreseeable future. According to recent estimates, most of the traffic carried today over Internet uses TCP (Transmission Control Protocol) as transport protocol (Kurose and Ross 2005). This figures the key rule that TCP plays in delivering a reliable service to the most common network applications such as email programs and Web browsers.

TCP has been subject to many performance studies based on simulations, measurements, and analytical models. Modelling the TCP behaviour using analytical paradigms is the key to obtain more general and parametric results and to achieve a better understanding of the TCP behaviour under different operating conditions. On the other hand, the development of accurate models of TCP is difficult, because of (1) the intrinsic complexity of the protocol algorithms, (2) the complex interactions between TCP and the underlying IP network.

Accurate modelling of individual TCP flows requires modelling of complex dynamics rising from the additive-increase/multiplicative-decrease (known as the congestion control) mechanism of TCP protocol, session dynamics, and heterogeneous round-trip delays in conjunction with the underlying network layer. As the size of state space explodes with the number of sessions, this represents a major obstacle to modelling the interaction of many TCP flows in a realistic setting. For the same reason, even numerical experiments become computationally prohibitive, and fail to provide an insight into the complex dynamics. The existing literature on TCP traffic modelling usually skirts these major obstacles by relying on ad hoc assumptions, which causes the model to be accurate only in certain regimes (Stevens 1994).

Extensive studies of TCP have provided detailed performance of the TCP and UDP of single TCP sessions in terms of their throughputs and delays. Several models have been also proposed for multiple TCP sessions. Multiple TCP sessions can be categorized to *persistent* and *non-persistent* connections (Kurose and Ross 2005). Non persistent models can be considered as a reasonable approximation for the Web traffic using HTTP. On the other hand, persistent models are useful for the files sent by e-mail or FTP servers. It can be noted the importance of both models. Few models have been proposed for non-persistent connections. However, the accuracy of these models in the real networks is limited by the assumptions made and the time scale at which one is looking at the performance. Moreover, most of the *non-persistent* models that have appeared in the literature so far consider the interaction among TCP connections that share one high-load link (called the bottleneck link), but assume independence among the behaviours of the TCP connections that share this link.

In this paper, our model presents a microscopic, i.e. showing the details of the TCP connection, scalable model for non-persistent TCP sessions that can be used to deeply understand how parameters, such as propagation delay, router's bottleneck buffers, and rate of variation of the available bottleneck bandwidth, affect the performance indices of the internet, such as the end-to-end throughput and delay. The model relaxes some of the limiting assumptions of other suggested models in the literature. We assume that traffic flows through a bottlenecks routers which has real time TCP ON–OFF traffic and a stochastic flow of non-persistent TCP connections going through it. Our analysis is able to handle any number of TCP connections and still provides accurate performance metrics of each of these connections. We can assume different packet sizes, forward and backward round trip times and max window size for each of these connections. The model represents *N* non-persistent TCP sessions. It is based on modelling one TCP session and the effect of the other sessions is included as a summed average number of segments implanted in the shared bottleneck channel. The model can be easily tailored to model also persistent TCP connections.

The rest of the paper is organized in six sections as follows: (1) Related work is described. (2) The congestion avoidance mechanisms are reviewed. (3) Model assumptions are discussed. (4) The suggested model is described. (5) The analytical and simulation results are presented. (6) The work is concluded.

## RELATED WORK

Researchers have proposed a number of different models, from detailed single flow models to predict the throughput of a single flow as a function of round-trip delay and packet loss probability (Altman et al. 2000; Ranjan et al. 2004; Padhye et al. 2000) to linear and nonlinear *macroscopic* models, i.e. dynamics of TCP are aggregated, motivated by limit theorems to derive limiting traffic models (Hollot et al. 2001; Mathis et al. 1997). As a result, the behaviour of a single long-lived TCP flow is relatively well understood.

Recently, several models of multiple TCP sessions passing through the Internet have been analyzed (Nicky et al. 2007; Altman et al. 2005; Sharma and Gupta 2006; Marsan et al. 2000). However, most of the studies on multiple TCP sessions assume a fixed number of *persistent TCP connections*, where it is assumed that there are a fixed number of TCP sessions which always have packets to send (Abouzeid et al. 2000; Brown 2000; Nicky et al. 2007; Altman et al. 2005). These models can be considered a reasonable approximation if the TCP sessions are sending long files (e.g. FTP sessions). However, most of the WAN TCP traffic today consists of Web traffic which contains small files. Several models (*non-persistent sessions*), where TCP sessions arrive, send a finite file and then leave have been studied (Tinnakornsrisuphap and La 2006; Sharma and Gupta 2006; Marsan et al. 2000). Models of *non-persistent* TCP connections usually utilize ON–OFF sources. It is to be noted that these models can be useful when an e-mail or FTP server is sending a small file.

In all the *non-persistent* studies, a single bottleneck queue is considered; there is no UDP traffic in the queue and the TCP parameters (propagation delays, maximum window size etc) are same for all the TCP connections. Furthermore, most of these studies (Tinnakornsrisuphap and La 2006; Sharma and Gupta 2006) have formulated *macroscopic* models (ignoring the packet level details, in particular the packet lengths). Few of them have been conducted using *microscopic* models (Marsan et al. 2000). However, assumptions used in *non-persistent* studies have a significant impact on the performance model of the TCP connections.

In (Marsan et al. 2000), Marsan *et al.* developed a scalable microscopic model for the behaviour of a number of non-persistent TCP connections that share a bottleneck link, considering the synchronization among connections. The model is based on Markovian assumptions and on two GSPN (generalized stochastic Petri net (Marsan et al. 1995)) descriptions of the system. The two models are called the multi-user GSPN model and the single-user GSPN model. The two models are iteratively solved, until convergence of the average window size values. The TCP source model incorporates both slow-start and congestion avoidance. From the two GSPN descriptions, using a fixed-point algorithm, several interesting performance metrics of the TCP connections was derived. The model was validated through extensive comparisons of the model's results with the output of the simulator "ns version 2" (VINIT Project 1997). In spite of its scalability, several assumptions were introduced. In Section 5, we will discuss several of these assumptions and show how our model outperforms Marsan's model. We consider this model the closest in spirit to our work due to its scalability and its microscopic view to the TCP behaviour hence we will compare our performance metrics to those obtained by (Marsan et al. 2000).

## THE CONGESTION CONTROL MECHANISM

The congestion control mechanism, described in RFC 2581, allows the TCP congestion window size (denoted by *cwnd*) to increase according to two different mechanisms: *slow start* (*SS*), where *cwnd* doubles every round-trip time, or *congestion avoidance* (*CA*), where *cwnd* increases by one every round-trip time. The congestion control mechanism also responds to three events namely: acknowledgment (*Ack*) reception, timeout and triple *Ack* reception. When the TCP connection is set up, the protocol starts in *SS* mode with *cwnd* equal to 1, and enters *CA* mode as soon as *cwnd* reaches a predefined threshold value (called *ssthresh*). The TCP window can grow up to a maximum allowable value that is denoted by $w_m$.

A timeout event allows the TCP source to detect segment losses. The timeout expiration (*RTO*) is computed as the average estimated round-trip time (*RTT*) plus four times the estimated round-trip time standard deviation. Whenever a segment loss is detected, the TCP threshold (*ssthresh*) is set to half the current window size, and the window size is set to 1 (i.e. *ssthresh* = *cwnd*/2 , *cwnd* = 1). Moreover, the TCP enters the *SS*. The congestion control mechanism can be clarified in the following algorithm, where MSS is the maximum segment size:

**Congestion Control Mechanism**

```
Begin at Init. Mode
• Initialization mode:
      cwnd = 1, ssthresh = 65,536 B
      Go to SS mode
• Slow Steady (SS) mode:
  - If Ack. Received cwnd = cwnd + MSS
  - If cwnd > ssthresh Go to CA model
  - If Timeout event
      ssthresh =cwnd/2, cwnd = 1 MSS
      Remain in SS mode
  - If Triple Ack. Received (TCP Reno)
        $ssthresh = cwnd/2$, $cwnd = ssthresh$
        Go to CA mode
• Congestion Avoidance (CA) mode:
  - If Ack. Received
        $cwnd = cwnd + MSS \bullet (MSS/cwnd)$
  - If Timeout event
        $ssthresh = cwnd/2$, $cwnd = 1\ MSS$
        Go to SS mode
  - If Triple Ack. Received (TCP Reno)
        $ssthresh = cwnd/2$, $cwnd = ssthresh$
        Remain in CA mode
```

## MODEL ASSUMPTIONS

We interested in studying the performance of TCP sources that their data packets pass through one of the most congested links in WAN and the acknowledgments of that packets pass through another congested link in the same WAN. It is important to study not only the effect of loosing data packets but also the acknowledgments packets, because they have the same effect in deterioration of the performance of TCP sources.

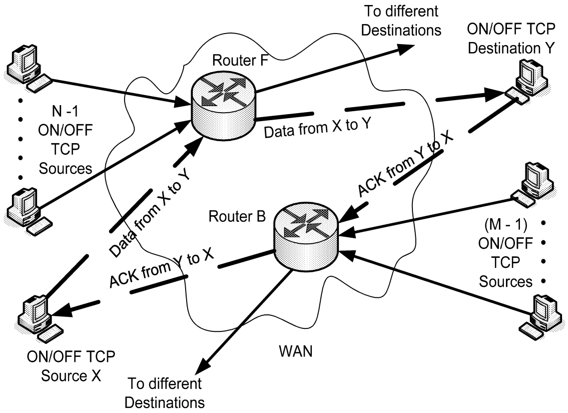


Figure 1: WAN with Forward and Backward Bottleneck

Figure 1 depicts the network model used in the following analysis and simulation. It consists of: (1) Wide area Network (WAN), (2) a TCP source called X that send data to a destination called Y on the other side of WAN, (3) (*N*-1) TCP source that send data to unknown destination in the other side of WAN, (4) (*M*-1) TCP sources that send data to unknown destination in the other side of WAN, (5) two routers, which are connected to the most congested links in WAN (called bottlenecks), router F (forward bottleneck) and router B (backward bottleneck). We supposed that all traffic of the *N* TCP sources (in the left of Figure 1) pass through the forward bottleneck and all traffic of the *M* TCP sources (in the right of Figure 1) pass through the backward bottleneck. Therefore, when the TCP source X send a data packet to the destination Y, the packet will pass through the forward bottleneck and the acknowledgment from Y to X will pass through the backward bottleneck.

We supposed that all sources use the web browsing applications in the application layer and utilize non-persistent TCP connections to send their data. Therefore, these sources use the TCP connections to send burst traffic during active periods (ON periods) of sources. When the web applications in these sources have no data to send, the TCP connection will be idle (OFF period). Hence, we call these sources ON/OFF TCP sources. Also we assumed that the time of ON period (*Ton*) and time of OFF period (*Toff*) of TCP connections are random variable with negative exponential distribution.

We have used SRN (stochastic reward net) for representing the complex behaviour of the congestion mechanism. SRNs are a variant of the GSPN (Marsan et al. 1995) which allow defining enabling and weight functions. Both the SRN and the GSPN are built on an underlying Markov process. The development of a model with Markovian characteristics requires the introduction of exponential assumptions for the time intervals of the model represented by transitions (denoted by $t_i$). Hence all transitions (SRN are built from transitions and places) listed in this section and the next section have an assumed exponential distribution. Moreover, the dynamics of the TCP window are governed by the TCP congestion control algorithms which are too complex to be exactly accounted for in an analytical model. Hence several assumptions were considered to elevate this complexity, which can be summarized in the following points:

- No fast retransmit/fast recovery mechanisms are considered. We will consider these mechanisms in future works.
- The TCP sources send a number of packets equal to *cwnd* in a burst, and the acknowledgments return in a burst. This mean that the congestion window size approximately changes after a period of time equal to one round trip time.
- The TCP window size (*cwnd*) isn't measured in bytes, as in real world networks, but it is measured using Maximum Segment Size (MSS), because it is more practical.
- The acknowledgments packets are piggybacked with data and their size equal the size of data packets.
- One of the features of TCP, that can be enabled or disabled, is the delayed Ack, which provides for sending one Ack for every two received packets. We didn't take into account such feature, although the proposed model can easily be modified to adopt this feature.
- The propagation delays of all sources in the forward direction are equal. Also, the propagation delays of all sources in the backward direction are equal.

In the timeout mechanism, whenever a segment loss is detected, the TCP threshold *ssthresh* is set to half the current window size, and the window size is set to 1. In (Marsan et al. 2000), the mechanism by which *ssthresh* (denoted as $W_t$ in (Marsan et al. 2000)) is adjusted is simplified. It was assumed that the only values which $W_t$ can take are 1 and $½w_m$ (i.e., the minimum and maximum allowed values of $W_t$ are considered). In our approach, no approximation was made with respect to the mechanism by which *ssthresh* is adjusted. Also in (Marsan et al. 2000) to simplify the model they considered that for every packet received by destination the source will receive the corresponding Ack packet after half of the RTT. In the proposed model we suppose that the Ack packets may have different queuing delays than data packets and may be lost in the way to the source. It can be seen in Section 5 that our model allows supposing forward propagation time different from the backward propagation time. Similarly the model has the flexibility to assume different forward and backward bottlenecks. Moreover, it allows modelling of

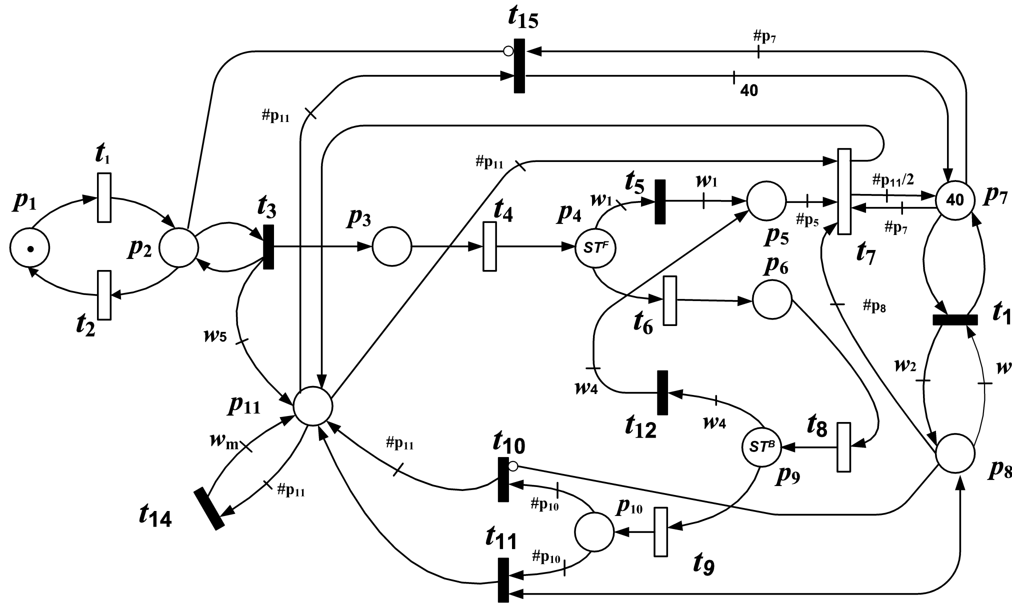


Figure 1: SRN Model of Non-Persistent TCP Sessions

*piggybacking* mode of TCP sessions where the *Acks* and data are transferred in the same segment.

## MODEL DESCRIPTION

In this section, we will give a description of the suggested non-persistent TCP source model. The model represents *N* non-persistent TCP sources. Figure 2 shows the suggested SRN model for non-persistent TCP sessions in an ON/OFF source through congestion control. It is based on modelling one TCP sessions in one ON/OFF source and the effect of the other sessions from other sources is included as a scaled or summed average number of segments implanted in the shared bottlenecks channel. Table 1 shows the arc weight functions of some arcs in the SRN shown in Figure 1, where # $P_i$, $Q^F$ and $Q^B$ are the number of tokens in the place $P_i$, the forward bottleneck buffer size, and the backward bottleneck buffer size, respectively.

Table 1: Arcs Weight Functions

| weight | Arc weight Function |
|---|---|
| $w_1$ | $\# P_4 - Q^F$ |
| $w_2$ | 1 if $\# P_8 = 0$<br>0 if $\# P_8 = 1$ |
| $w_3$ | $\# P_8$ |
| $w_4$ | $\# P_9 - Q^B$ |
| $w_5$ | 1 if $\# P_{11} = 0$<br>0 if $\# P_{11} = 1$ |

Tokens in places $P_1$ and $P_2$ represent Off and On states of the TCP source respectively. The timed transitions $t_1$ and $t_2$ model the status change of the TCP source states. If *Ton* and *Toff* are the average ON and OFF time of TCP sources, thus the rates of $t_1$ and $t_2$ are 1/*T*on and 1/*T*off respectively. During ON periods, the TCP source is allowed to transmit segments according to the evolution of their TCP window. The number of tokens in place $P_{11}$ represents the current window size (*cwnd*). In the beginning $P_{11}$ is empty. Hence when the TCP source is in ON state (token in place $P_2$), transition $t_3$ is enabled. When the first TCP source becomes active, transition $t_3$ puts one token in $P_{11}$ which starts the TCP window growth mechanism (*cwnd* =1).

Firing of $t_3$ places a token in $P_3$, that represents a temporary buffer for sent segments. It is assumed that all segments are sent in a short burst. Hence, accumulation of tokens in $P_3$ continues until it reaches $\#P_{11}$ (*cwnd*), where $\#P_{11}$ represents the number of token in $P_{11}$. Forward propagation delay ($D_P^F$) encountered by segments is represented by $t_4$, where its rate is evaluated by $1/D_P^F$. In our model, we have differentiated between forward and backward propagation delays. The bottleneck buffer is represented by $P_4$ which allows for modelling the transaction between several TCP sessions.

Segments (tokens in $P_4$) are subjected either to be correctly acknowledged or lost due to buffer overflow. Correct acknowledgment is handled by $t_6$ which model transmission of a packet from the TCP source. The rate of $t_6$ is calculated by $B^F/S_P$, where $B^F$and $S_P$ represent the bandwidth of the forward bottleneck link and the packet size. The acknowledgment segment passes through $P_6$, $t_8$, $P_9$, $t_9$ in turn until it reaches $P_{10}$, where the acknowledgment segments are accumulated. Discussion of the function of $P_6$, $t_8$, $P_9$, $t_9$ is similar to the previous discussion of $P_3$, $t_4$, $P_4$, $t_6$ except that $D_P^F$ and $B^F$ are replaced by $D_P^B$ and $B^B$ , where $D_P^B$ and $B^B$ represent the backward propagation delay and bandwidth of the backward bottleneck link. It can be noted that any performance for the acknowledgment segments can be fulfilled especially if it is piggybacked by data.

Whenever a packet loss occurs (firing of transition $t_5$), one token is put in place $P_5$, whose marking represents the occurrence of at least one segment loss. The presence of a token in $P_5$ enables the timed transition $t_7$, which models the timeout expiration and loss detection. Transition $t_7$ has rate equal to 1/*Tout*. The TCP protocol sets the timeout *Tout* based on the estimation of the average or Round Trip Time (RTT). RTT is the time required for a packet to travel from source to destination and its Ack return back to the source. RTT can be computed as follows:

$$\text{RTT} = D_P^F + D_Q^F + D_T^F + D_P^B + D_Q^B + D_T^B$$

where $D_Q^F, D_Q^B$ , $D_T^F$ and $D_T^B$ are the queuing delay in the forward bottleneck buffer, the queuing delay in the backward bottleneck buffer, the transmission time in the forward direction and the transmission time in the backward direction, respectively. $D_T^F$ and $D_T^F$ are neglected because they are very small compared to propagation and queuing delay. Because the RTT is a random variable, the TCP compute the timeout as follows: $Tout = A_{RTT} + 4\sigma$, where $A_{RTT}$ and σ are the average estimated round trip time and the round trip time standard deviation. The round trip time is modelled as an exponential random variable, therefore $A_{RTT} = \sigma$, consequently $Tout = 5\,A_{RTT}$.

Firing of $t_7$ causes halving the value of # $P_7$ which represents the value of *ssthresh*. It is to be noted that we have chosen a value of "40" which represents the initial value of packets in *ssthresh*. This value comes from the fact that max value of bytes in *ssthresh* is 65536 and the average size of a packet is "1500" (40 = 65536/1500). Moreover firing of transition $t_7$ represents the loss condition which causes putting a one in *cwnd* place ($P_{11}$).

The marking of the place $P_8$ indicates the mode of growth of the TCP window (either slow start or congestion avoidance). Since only one mode of window growth is possible at a time. Presence of a token in $P_8$ indicates that it is in *CA*, otherwise, it is in the *SS*. When there is a token in place $P_{10}$ and no tokens in $P_8$, the *SS* algorithm is initiated. Firing of $t_{13}$ occurs when the condition (*cwnd* > *ssthresh*) (or $\#P_{11} > \#P_7$) is fulfilled. This is in turn causes change in the congestion mode of the TCP session. Switching from *SS* to *CA* is done through $t_{13}$ which places a token in $P_8$. In addition when the timeout is detected, transition $t_7$ removes tokens from $P_8$, consequently the mode is switched from CA to SS. Transition $t_{10}$ represents the activity of *SS* by incrementing exponentially *cwnd* through doubling tokens in $P_{11}$ per *RTT*. Similarly transition $t_{11}$ represents the activation of *CA* by incrementing linearly *cwnd* (tokens in $P_{11}$) by one unit per *RTT*.

Transition $t_{14}$ prevents the number of tokens in $P_{11}$ to increase over than the maximum window size $w_m$. Transition $t_{15}$ is enabled if the session ends (no tokens in $P_2$). Firing of $t_{15}$ causes flushing of $P_{11}$ and initializing of $P_7$ with "40". Flow control (GO back N) is fulfilled through the enabling rule of $t_3$ , where it is only enabled if acknowledgment is received for sent segments.

All TCP sources in the forward (or backward) direction are sharing the forward (or backward) bottleneck buffer size and link capacity. Therefore, in a TCP source model to model the effect of other TCP sources ((*N*-1) sources) we put stationary tokens (ST) in the forward (and backward) buffer which represent mean number of packets that all other source will send to the bottleneck buffer at any time. ST is a new type of tokens in Petri nets that don't move to any other place. They can easily be configured by controlling the firing and enabling rules of transitions that their firing move ST to another place. The Number of tokens in forward and backward buffer bottleneck at any time and ST can be computed as follows:

$$ST^F = \sum_{i=1}^{N-1} A_i \qquad ST^B = \sum_{i=1}^{M-1} A_i \qquad (1)$$

$$\#\,P_4 = \#\,P_{11} + ST^F \qquad \#\,P_9 = \#\,P_{11} + ST^B \qquad (2)$$

where $\boldsymbol{ST^F}$, $\boldsymbol{ST^B}$, and $\boldsymbol{A_i}$ represent the ST in the forward bottleneck, ST in the backward bottleneck, and the average number of tokens in place $P_4$ for all sessions of the TCP source *i*. It is clear that if we assume that all TCP sources have the same parameters (propagation delay, maximum window size, packet size, ON/OFF time), equation 1 is modified to be

$$ST^F = N\bullet A \qquad ST^B = M\bullet A \qquad (3)$$

To reduce the computational efforts to compute ST when the TCP sources have different parameters, we compute the average value of these parameters which are used to solve the SRN model to compute *A*. After that, equation (3) is used to compute the number of ST. Tables 2, 3 and 4 list the rates, priorities and enabling rules of the timed and immediate transitions of the proposed model shown in Figure 2.

Table 2: Transitions rates

| Transition | Rate |
|---|---|
| $t_1$ | $1/Ton$ |
| $t_2$ | $1/Toff$ |
| $t_4$ | $1/D_P^F$ |
| $t_6$ | $1/D_Q^F = B^F/S_P$ |
| $t_7$ | $1/Tout$ |
| $t_8$ | $1/D_P^B$ |
| $t_9$ | $1/D_Q^B = B^B/S_P$ |

Table 3: Transitions priorities

| Transition | Priority |
|---|---|
| $t_3$ | 1 |
| $t_5$ | 1 |
| $t_{10}$ | 2 |
| $t_{11}$ | 2 |
| $t_{12}$ | 1 |
| $t_{13}$ | 1 |
| $t_{14}$ | 2 |
| $t_{15}$ | 3 |

Table 4: Transitions enabling rules

| Transition | Enabling Rule |
|---|---|
| $t_3$ | $(\#P_3 + \#P_4 + \#P_5 + \#P_6 + \#P_9 + \#P_{10} - \text{ST}^F - ST^B) < \#P_{11}$ |
| $t_5$ | $\#P_5 > Q$ |
| $t_6$ | $\#P_4 > ST^F$ |
| $t_7$ | $(\#P_5 \geq 1) \wedge (\#P_{11} \geq 1) \wedge (\#P_3 + \#P_4 + \#P_6 + \#P_9 - ST^F - ST^B) = 0$ |
| $t_{10}$ | $\#P_{10} \geq \#P_{11}$ |
| $t_{11}$ | $\#P_{10} \geq \#P_{11}$ |
| $t_{13}$ | $(\#P_{11} > \#P_7) \wedge (\#\,P_8 = 0)$ |
| $t_{14}$ | $\#P_{11} > w_m$ |

## RESULTS

In this section, we examine the accuracy of our proposed model by making extensive comparisons of its results with the results of simulation experiments. The simulation results were obtained by using ns-2 simulator (VINIT Project 1997). The ns2 is one of the most powerful tools for extracting accurate performance indices for different algorithms of TCP protocol. Also, it provides quite detailed descriptions of the dynamics of other Internet protocols.

Two fundamental measures are used to evaluate the proposed SRN model and compare it with simulation of TCP Tahoe, including all of its features: average segments loss probability and the average TCP connections throughput. The throughput is average number of packets received and acknowledge by TCP destination. The segments loss probability and average throughput can be calculated from SRN model using the following equations

$$\text{Segments loss probability} = Pr\ (\#\ P_5 > 0)$$
$$\text{Average throughput} = \frac{A_W}{D_P^F + D_P^B + D_Q^F + D_Q^B}$$

Where $D_Q^F = Th(t_6)/M(P_4)$, $D_Q^B = Th(t_9)/M(P_9)$ and $Aw = M(P_{11})$. $A_w$ is the average congestion window size, $Th(t_6)$ and $Th(t_9)$ are the throughput of transitions $t_6$ and $t_9$ respectively, $M(P_4)$ and $M(P_9)$ are the mean number of tokens in places $P_4$ and $P_9$ respectively, and $Pr\ (\#\ P_5 > 0)$ is the probability that the number of tokens in place $P_5$ is greater than 0.

The simulated scenario consists of two sets of sources, one set in each direction, as shown in Figure 1. Each set of sources is either 10 or 20 non-homogeneous On/Off sources and multiplexed over a different bottleneck link. The packet size was assumed to be constant, equal to 1500 bytes, for all TCP connection. Also, the acknowledgment packet size was assumed to be 1500 byte, because we supposed that the acknowledgments carry data from destinations. The maximum congestion window size was considered to be 21 packets, corresponding to 32 KB, for all TCP connections. The bottlenecks buffer capacities of 50 and 100 packets were considered. The drop tail dropping policy was run in the bottlenecks buffer. We considered the cases of the bottleneck link capacity equal to either 100 or 400 Mb/s. All simulations were run until the width of the 98% confidence intervals fell within 2% of each throughput estimate. In all figures, 3–9, solid lines refer to simulation results, while dashed lines represents SRN model results.

We first consider a scenario with 10 homogeneous ON/OFF sources in the forward direction and the same number in the backward direction, non-persistent TCP connections with forward propagation delay equal to the backward propagation delay, bottleneck buffer size in the forward and backward direction equal to 50 packets, the bandwidth of the bottleneck links in the forward and backward direction equal to 100 Mbps, the average ON period of all sources are 200 ms, the average OFF period of all sources equal either 200 or 400 ms. In Figures 3 and 4, the throughput and segments loss probability were plotted versus increasing value of the one-way propagation delay. To illustrate the effect of OFF period of TCP sources on the throughput and segments loss probability, we considered two cases. The first one, labelled “Sim Toff = 200” in the figures, the average OFF period equal to 200 ms; in the second case, labelled “Sim Toff = 400”, the average OFF period equal to 400 ms.

As shown in Figures 3 and 4, when the propagation delay increases the throughput and the segments loss probability decreases because increasing propagation delay means that the distance between transmitter and receiver increases which makes the dynamic growth of the window slow down consequently reducing the generated traffic. We can notice that the average throughput and segments loss probability decreases with increasing the average OFF period of TCP sources, since the increasing of OFF period of TCP sources means that the sources are inactive for a longer period, consequently decreases the traffic from sources to destinations. In Figures 3 and 4, it is clear that the analytical model captures the behaviour of each source, providing very accurate predictions, as can be seen from the comparison with simulation results. Also, we can notice that the accuracy of the analytical results increases with increasing the propagation delay because the effect of queuing delay in computing *RTT* decreases, so the estimated error of the average queuing delay becomes less influent.

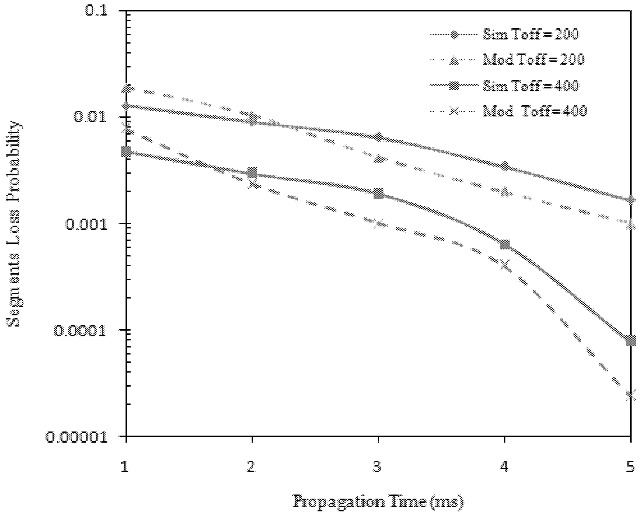


Figure 3: Segments Loss Probability for Different Values of One Way Propagation Delay, where  =*M* =10, $Q^F$ =$Q^B$ = 50, $B^F$=$B^B$= 100 Mbps, *Ton* =200 ms, *Toff* = 200 and 400 ms.

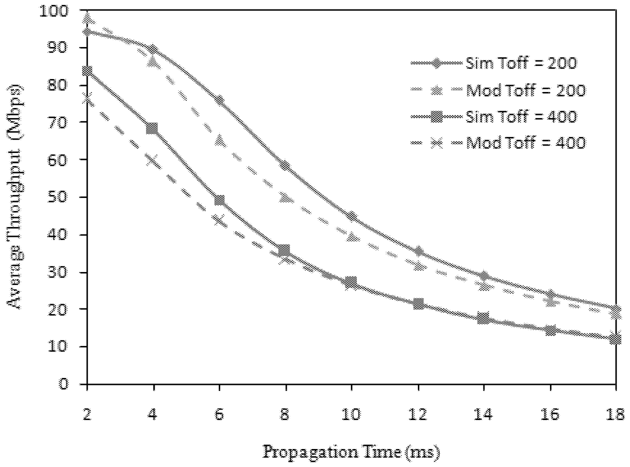


Figure 4: Average Throughput for Different Values of One Way Propagation Delay, where *N* =*M* = 10, $Q^F$= $Q^B$ = 50, $B^F$ =$B^B$ =100 Mbps, *Ton* =200 ms, *Toff* = 200 and 400 ms.

To investigate the effect of the number of TCP connections on the performance of the TCP sources, we set the bottleneck buffer size in the forward and backward direction equal to 50 packets, the bandwidth of the bottleneck link in the forward and backward direction equal to 100 Mbps, the average ON period of all sources are 200 ms, the average OFF period of all sources equal to 200 ms, the number of TCP connections equal either 10 or 20. We plot the segments loss probability and average throughput, in Figures 5 and 6 respectively, versus one way propagation delay, where the forward and backward propagation delays are equal. As expected, the packet loss probability and average throughput increase with increasing the number of TCP sources. Again, we can notice the accuracy of analytical performance predications compared to simulation is very high, especial for longer propagation delay. As clear in Figures 5 and 6, the accuracy of the analytical results didn't affect much by increasing the number of TCP sources. This means that the proposed method for modelling the correlation between TCP sources is very accurate and doesn't affect by the number of TCP sources. One of the main advantages of our model is that the solution time of the model doesn't increase much with increasing number of TCP sources, whereas the ns2 simulation time increases significantly with the increase in the number of TCP sources.

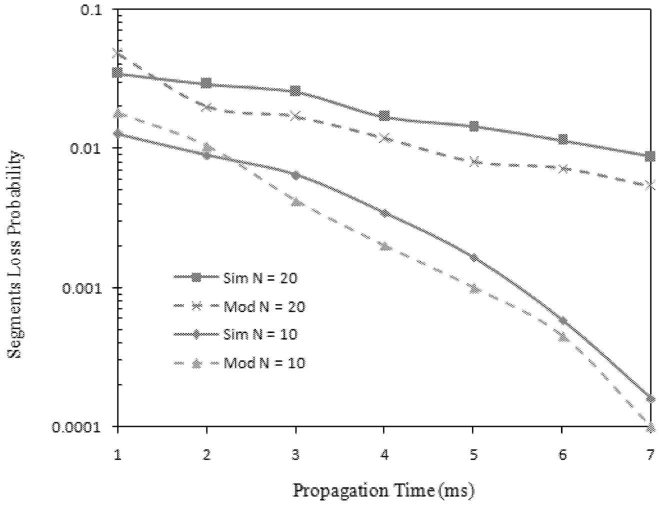


Figure 5: Segments Loss Probability for Different Values of One Way Propagation Delay, where $N = M = 10$ and 20, $Q^F = Q^B = 50$, $B^F = B^B = 100$ Mbps, $Ton = Toff = 200$ ms.

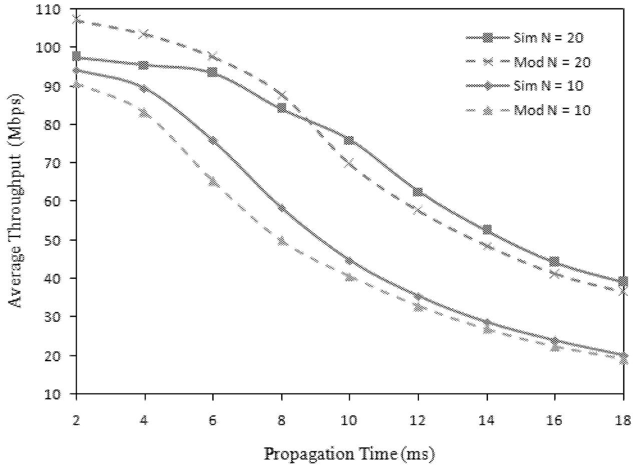


Figure 6: Average Throughput for Different Values of One Way Propagation Delay, where $N = M = 10$ and 20, $Q^F = Q^B = 50$, $B^F = B^B = 100$ Mbps, $Ton = Toff = 200$ ms.

To assess the sensitivity of the accuracy of the proposed model with changing the buffer size, in Figures 7 and 8 we plotted the segments loss probability and average throughput verses one way propagation delay where the forward and backward buffer size equal either 50 or 100 packets; the bandwidth of the bottleneck link in the forward and backward direction equal to 100 Mbps, both ON and OFF periods have an average value equals 200 ms, the number of forward or backward TCP sources equal 10. As shown in Figures 7 and 8, it is clear that the increasing of the buffer size decreased the packet loss probability and increased the average throughput, as expected. Decreasing the segments loss probability reduce the probability of the sudden decreasing of congestion window size to be equal one after timeout, which let sources generate more packets that pass through the network, consequently the throughput increases. Also in this scenario, the proposed model results are accurate compared to the simulation results.

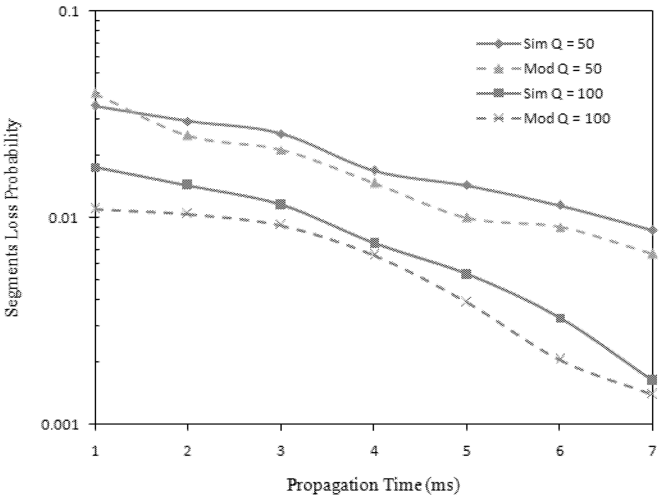


Figure 7: Segments Loss Probability for Different Values of One Way Propagation Delay, where $N = M = 10$, $Q^F = Q^B = 50$ and 100, $B^F = B^B = 100$ Mbps, $Ton = Toff = 200$ ms.

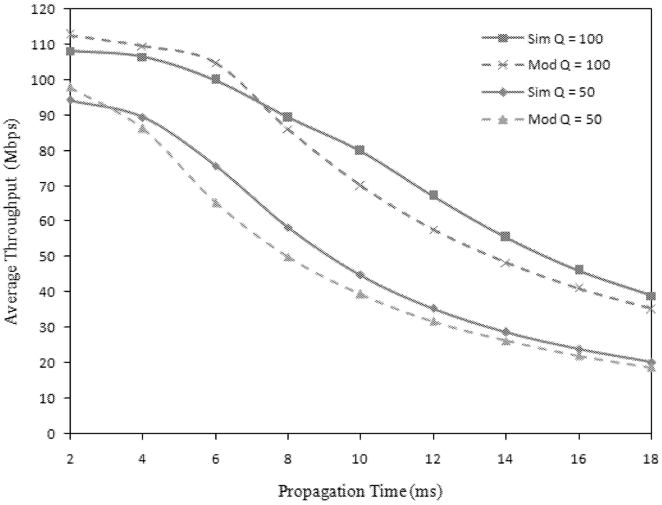


Figure 8: Average Throughput for Different Values of One Way Propagation Delay, where $N = M = 10$, $Q^F = Q^B = 50$ and 100, $B^F = B^B = 100$ Mbps, $Ton = Toff = 200$ ms.

The model accuracy for different number of TCP connections can be further discussed by observing the results in Tables 5 and 6, which report the relative errors in average throughput and segments loss probability for the proposed model and Marsan's model (Marsan et al. 2000) for different values of one way propagation delay. The model parameter were as follows: the buffer size equals 50 segments, the average ON period of TCP sources equals 200 ms, the average OFF period of TCP sources equals 200 ms, the bandwidth of bottleneck link equals 100 Mbps, the number of TCP sources equals either 10 or 20. The relative percentage error was computed using the following equation

$$\% \, Error = \frac{Simulation - Modeling}{Simulation} \times 100$$

Where *Simulation* and *Modelling* are the value of average throughput or the segments loss probability computed using simulation and analytical modelling, respectively. It can be noted from Tables 5 and 6 that the percentage errors in the estimation of the average TCP connections throughput always remain within an absolute band of roughly 10% compared to 20% for Marsan. Also errors in the estimation of the TCP segments loss probability always remain within a band of 40 %, while that of Marsan can grow quite high, particularly in the case of a lower number of TCP connections.

Table 5: Relative Percentage Error of Loss Probability

| $D_P$ | N = 10 | | N = 20 | |
|---|---|---|---|---|
| | Marsan's Model | Proposed Model | Marsan's Model | Proposed Model |
| 1 | - 1511 | - 40.8 | - 321 | - 37.9 |
| 2 | - 2720 | - 15.9 | - 511 | 31.6 |
| 3 | - 432 | 34.8 | - 702 | 33.2 |
| 4 | - 250 | 41.6 | - 316 | 30.3 |
| 5 | - 123 | 39.4 | -217 | 44.2 |

Table 6: Relative Percentage Error of Throughput

| $D_P$ | N = 10 | | N = 20 | |
|---|---|---|---|---|
| | Marsan's Model | Proposed Model | Marsan's Model | Proposed Model |
| 2 | - 21.1 | 3.7 | - 19.6 | - 9.7 |
| 4 | - 14.2 | 6.8 | - 7.5 | - 8.2 |
| 6 | - 6.3 | 13.7 | 18.8 | - 4.3 |
| 8 | 18.6 | 14.4 | 17.9 | - 4.3 |
| 10 | 17.7 | 9.4 | 16.2 | 8.2 |
| 12 | 15.3 | 7.4 | 12.7 | 7.9 |
| 14 | 14.9 | 6.5 | 11.4 | 7.5 |
| 16 | 14.4 | 6.2 | 11.1 | 7.0 |
| 18 | 13.9 | 5.3 | 10.6 | 6.2 |

In summary, after testing the proposed model for different values of all parameters that govern the model such as the buffer size, the number of TCP sources, and propagation delay, we found that the model is sensitive to any change in any of these parameters and provide a very accurate prediction of the average throughput compared to simulation. Although the prediction of segment loss probabilities is intrinsically more critical, nevertheless the proposed modelling approach still succeeds in providing reasonable accuracies.

## CONCLUSIONS

In this paper, we have presented a microscopic scalable model for non-persistent TCP sessions. The model relaxes some of the limiting assumptions of other suggested models in the literature. We assume bottleneck routers which have real time TCP ON–OFF traffic and a stochastic flow of non-persistent TCP connections going through them. Our analysis is able to handle any number of TCP connections and still provides accurate performance metrics of each of these connections. We can assume different parameters, such as ON/OFF time and forward and backward round trip times for each of these connections.

The model represents *N* non-persistent TCP sessions. It is based on modelling one TCP session using SRN and the effect of the other sessions is included as a summed average number of segments implanted in the shared bottlenecks channel. Several interesting performance metrics referring to the TCP connections can be derived. A careful validation of our modelling approach was conducted through extensive comparisons of the analytical results with the outputs of ns-2 "*ns* version 2". Comparisons showed that our model succeeds in providing an accurate representation of the behaviour of TCP connections under several different settings. A comparison with a scalable model of the literature was also drawn to validate the efficiency of our model.

## REFERENCES

Abouzeid, A.A.; S. Roy; and M. Azizoglu. 2000. "Stochastic modelling of TCP over lossy links." P*roceedings of the IEEE INFOCOM.*

Altman, E.; K. Avrachenkov; and C. Barakat. 2000. "TCP in presence of bursty losses." in *Proc. ACM SIGMETRICS*, Santa Clara, CA, pp. 124–133.

Altman, E.; K. Avrachenkov; and C. Barakat. 2005. "A stochastic model of TCP/IP with stationary random losses." *IEEE/ACM Transactions on Networking (TON)*, Vol. 13, Issue: 2, April 1, pp. 356-369.

Brown, P. 2000. "Resource sharing in TCP connections with different round trip times." *IEEE INFOCOM.*

Hollot, C. V.; V. Misra; D. Towsley; and W. B. Gong. 2001. "A control theoretic analysis of RED." in *Proc. IEEE INFOCOM*, pp. 1510–1519.

Kurose, J. F. And K. W. Ross. 2005. "Computer Networking: A Top-Down Approach Featuring the Internet." *Addison Wesley*, 3rd Edition.

Marsan, M.; G. Balbo; G. Conte; S. Donatelli; and G. Franceschinis. 1995. "Modelling with Generalized Stochastic Petri Nets." *Wiley*, New York.

Marsan, M.; C. Casetti, R. Gaeta; and R. M. Moe. 2000. "Performance analysis of TCP connections sharing a congested Internet link." *Performance Evaluation*, Vol. 42, Issue: 2-3, September 29, pp. 109-127.

Mathis, M.; J. Semske; J. Mahdavi; and T. Ott. 1997. "The macroscopic behaviour of TCP congestion avoidance algorithm." *Comput. Commun. Rev.*, vol. 27, no. 3, pp. 67–82,.

Nicky, D.; B. Haverkort, M. R. Mandjes; and W. R. Scheinhardt. 2007. "Versatile stochastic models for networks with asymmetric TCP Sources." *Performance Evaluation*, Vol. 64, Issue: 6, pp. 507-523.

Padhye, J.; V. Firoiu; D. Towsley; and J. Kurose. 2000. "Modelling TCP Reno performance: a simple model and its empirical validation." *IEEE/ACM Trans. Netw.*, Vol. 8, Issue: 2, pp. 133–145.

Ranjan, P.; E. H. Abed; and R. J. La. 2004. "Nonlinear instabilities in TCPRED." *IEEE/ACM Trans. Netw.*, vol. 12, no. 6, pp. 1079–1092.

Sharma,V.; and A. Gupta. 2006. "A Unified Approach for Analyzing Persistent, Non-Persistent and ON–OFF TCP sessions in the Internet." *Performance evaluation*, Vol. 63, Issue: 2, pp. 79-98.

Stevens R. 1994. "TCP/IP illustrated, volume1: The protocols." *Addison Wesley.*

Tinnakornsrisuphap, P.; R. J. La. 2006. "Asymptotic behaviour of heterogeneous TCP flows and RED gateway." *IEEE/ACM Transactions on Networking (TON)*, Vol. 14, Issue: 1, February 1, pp. 108-120.

VINIT Project in http://www.isi.edu/nsnam/vint/index.html

# A Simulation Model for Evaluating Distributed Systems Dependability

Ciprian Dobre, Florin Pop, Valentin Cristea
*Faculty of Automatics and Computer Science, University Politehnica of Bucharest, Romania*
E-mails: {ciprian.dobre, florin.pop, valentin.cristea}@cs.pub.ro

## Abstract

*In this paper we present a new simulation model designed to evaluate the dependability in distributed systems. This model extends the MONARC simulation model with new capabilities for capturing reliability, safety, availability, security, and maintainability requirements. The model has been implemented as an extension of the multithreaded, process oriented simulator MONARC, which allows the realistic simulation of a wide-range of distributed system technologies, with respect to their specific components and characteristics. The extended simulation model includes the necessary components to inject various failure events, and provides the mechanisms to evaluate different strategies for replication, redundancy procedures, and security enforcement mechanisms, as well. The results obtained in simulation experiments presented in this paper probe that the use of discrete-event simulators, such as MONARC, in the design and development of distributed systems is appealing due to their efficiency and scalability.*

**Keywords**: Distributed Systems, Grid Computing, Modeling and Simulation, Dependability Model, Performance Analysis.

## 1. Introduction

Modeling and simulation were seen for long time as viable solutions to develop new algorithms and technologies and to enable the enhancement of large-scale distributed systems, where analytical validations are prohibited by the scale of the encountered problems. The use of discrete-event simulators in the design and development of large scale distributed systems is appealing due to their efficiency and scalability.

Together with the extension of the application domains, new requirements have emerged for large scale distributed systems; among these requirements, reliability, safety, availability, security and maintainability, in other words dependability [1], are needed by more and more modern distributed applications, not only by the critical ones.

However, building dependable distributed systems is one of the most challenging research activities. The characteristics of distributed systems make dependability a difficult problem from several points of view. The geographical distribution of resources and users that implies frequent remote operations and data transfers lead to a decrease in the system's safety and reliability and make it more vulnerable from the security point of view. Another problem is the volatility of the resources, which are usually available only for limited periods of time. The system must ensure the correct and complete execution of the applications even in the situations when the resources are introduced and removed dynamically, or when they are damaged. The management of distributed systems is also complicated by the constraints that the applications and the owners of the resources impose; in many cases there are conflicts between these constraints – for example, an application needs a long execution time and performs database operations, while the owner of the machine on which the application could be run only makes it available in a restricted time interval and does not allow database operations.

In this paper we present a simulation model designed to evaluate the dependability in distributed systems. The proposed model extends the MONARC simulation model [16] with new capabilities for capturing reliability, safety, availability, security, and maintainability requirements. The model has been implemented as an extension of the multithreaded, process oriented simulator MONARC, which allows the realistic simulation of a wide-range of distributed system technologies, with respect to their specific components and characteristics. The extended simulation model includes the necessary components to inject various failure events, and provides the mechanisms to evaluate different strategies for replication, redundancy procedures, and security enforcement mechanisms, as well. The paper extends the results presented in [10], introducing the simulation model designed for dependability of distributed systems. The results obtained in simulation experiments presented in this paper probe that the use of discrete-event simulators, such as MONARC, in the design and development of distributed systems is appealing due to their efficiency and scalability.

The rest of this paper is structured as follows. Section 2 presents related work in the field of modeling distributed systems, with a special accent on the evaluation of dependability. Next we present the MONARC architecture. The next sections present the simulation model being proposed, together with its implementation within the MONARC simulator. In section 6 we present the obtained results. Finally, in section 7 we present some conclusions and future work.

## 2. Related work

*SimGrid* [2] is a simulation toolkit that provides core functionalities for the evaluation of scheduling algorithms in distributed applications in a heterogeneous, computational Grid environment. It aims at providing the right model and level of abstraction for studying Grid-based scheduling algorithms and generates correct and accurate simulation results. *GridSim* [3] is a grid simulation toolkit developed to investigate effective resource allocation techniques based on computational economy. *OptorSim* [4] is a Data Grid simulator project designed specifically for testing various optimization technologies to access data in Grid environments. OptorSim adopts a Grid structure based on a simplification of the architecture proposed by the EU DataGrid project. *ChicagoSim* [5] is a simulator designed to investigate scheduling strategies in conjunction with data location. It is designed to investigate scheduling strategies in conjunction with data location.

None of these projects present general solutions to modeling dependability technologies for large scale distributed systems. They tend to focus on providing evaluation methods for the traditional research in this domain, which up until recently targeted the development of functional infrastructures. However, lately, the importance of dependable distributed systems was widely recognized and this is demonstrated by the large number of research projects initiated in this domain. Our solution aims to provide the means to evaluate a wide-range of solutions for dependability in case of large scale distributed systems.

Security in particular has never been properly handled by any of these projects before. The only currently existing simulator that offers the possibility to evaluate security solutions designed for distributed systems is G3S (Grid Security Services Simulator) [6]. It was developed so as to support various authentication mechanisms including X.509 certificates or Kerberos tickets and includes mechanisms for disseminating security threats, for evaluating various access control policies, etc. The simulator is based on the simulation model found in GridSim. We too support all the mechanisms found in G3S and some others. In addition we offer the possibility of evaluating security in a more general context, considering the entire context of distributed systems, with its specific characteristics.

An issue here is related to the generic evaluation of dependable distributed systems. A fault occurring in such systems could lead to abnormal behavior of any of the system's components. For this reason we argue that a correct evaluation of dependability in distributed systems should provide a complete state of the entire distributed system. Because of the complexity of the Grid systems, involving many resources and many jobs being concurrently executed in heterogeneous environments, there are not many simulation tools to address the general problem of Grid computing. The simulation instruments tend to narrow the range of simulation scenarios to specific subjects, such as scheduling or data replication.  The simulation model provided by MONARC is more generic that others, as demonstrated in [7]. It is able to describe various actual distributed system technologies, and provides the mechanisms to describe concurrent network traffic, to evaluate different strategies in data replication, and to analyze job scheduling procedures.

## 3. MONARC Architecture

MONARC is built based on a process oriented approach for discrete event simulation, which is well suited to describe concurrent running programs, network traffic as well as all the stochastic arrival patterns, specific for such type of simulations [8]. Threaded objects or "Active Objects" (having an execution thread, program counter, stack...) allow a natural way to map the specific behavior of distributed data processing into the simulation program. However, as demonstrated in [9], because of the considered optimizations, the threaded implementation of the simulator can be used to experiment with scenarios consisting of thousands of processing nodes executing a large number of concurrent jobs or with thousands of network transfers happening simultaneously.

In order to provide a realistic simulation, all the components of the system and their interactions were abstracted. The chosen model is equivalent to the simulated system in all the important aspects. A first set of components was created for describing the physical resources of the distributed system under simulation. The largest one is the regional center (Figure 1), which contains a site of processing nodes (CPU units), database servers and mass storage units, as well as one or more local and wide area networks. Another set of components model the behavior of the applications and their interaction with users. Such components are the "Users" or "Activity" objects which are used to generate data processing jobs based on different scenarios.

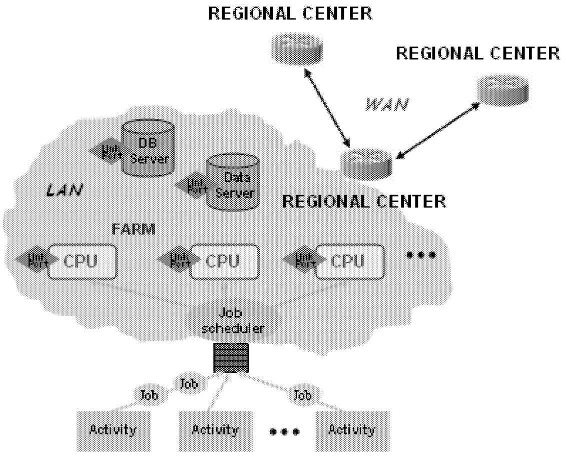


*Figure 1. The Regional center model.*

The job is another basic component, simulated with the aid of an active object, and scheduled for execution on a CPU unit by a "Job Scheduler" object. Any regional center can dynamically instantiate a set of users or activity objects, which are used to generate data processing jobs based on different simulation scenarios. Inside a regional center different job scheduling policies may be used to distribute jobs to corresponding processing nodes. One of the strengths of MONARC is that it can be easily extended, even by users, and this is made possible by its layered structure. The first two layers contain the core of the simulator (called the "simulation engine") and models for the basic components of a distributed system (CPU units, jobs, databases, networks, job schedulers etc.); these are the fixed parts on top of which some particular components (specific for the simulated systems) can be built. The particular components can be different types of jobs, job schedulers with specific scheduling algorithms or database servers that support data replication. The diagram in Figure 2 presents the MONARC layers and the way they interact with a monitoring system. In fact, one other advantage that MONARC have over other existing simulation instruments covering the same domain is that the modeling experiments can use real-world data collected by a monitoring instrument such as MonALISA, an aspect demonstrated in [11]. This is useful for example when designing experiments that are meant to experiment new conditions starting from existing real distributed infrastructures.

Using this structure it is possible to build a wide range of models, from the very centralized to the distributed system models, with an almost arbitrary level of complexity (multiple regional centers, each with different hardware configuration and possibly different sets of replicated data).

The maturity of the simulation model was demonstrated in previous work. For example, a number of data replications experiments were conducted in [8], presenting important results for the future LHC experiments, which will produce more than 1 PB of data per experiment and year, data that needs to be then processed. A series of scheduling simulation experiments were presented in [8], and [12].

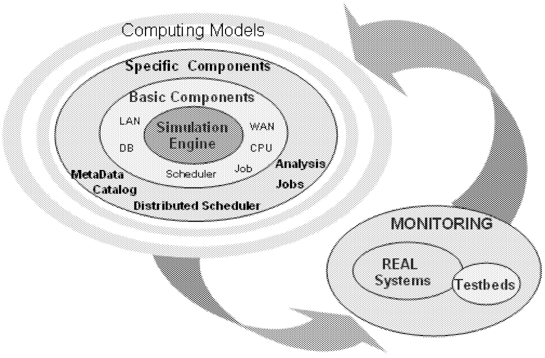


*Figure 2. The layers of MONARC.*

In [10] we presented an extension to the model designed to simulating fault tolerance in distributed systems using MONARC. The solution was able to model failures in distributed systems at hardware level (abnormalities in the functionality of hardware components) or software level (the middleware or application deviating from their normal functionality or delivery of services). In this we extended on this, implementing the proposed model in MONARC, evaluating it, but also adding additional mechanisms (security, different types of failures, at hardware and software levels, to model their occurrences and detection, as well as recovery and masking mechanisms) to cover a complete set of dependability characteristics, in its generic sense.

The characteristics of large scale distributed systems make the problem of assuring dependability a difficult issue because of several aspects. A first aspect is the geographical distribution of resources and users that implies frequent remote operations and data transfers; these lead to a decrease in the system's safety and reliability and make it more vulnerable from the security point of view. Another problem is the volatility of the resources, which are usually available only for limited periods of time; the system must ensure the correct and complete execution of the applications even in situations such as when the resources are introduced and removed dynamically, or when they are damaged.

In [10] we proposed an extension to the MONARC's model. In this paper we present the complete model designed to consider all aspects of dependability. Figure 3 presents the components of the dependable modeling layer. The extension to the simulation model relates to modeling faults appearing inside the modeled distributed system. In a distributed system failures can be produced at hardware level (abnormalities in the functionality of hardware components) or software level (the middleware or application deviating from their normal functionality or delivery of services). The simulation model accounts for both hardware, as well as software failures, modeling their occurrences and detection, as well as recovery and masking (redundancy) mechanisms [1].

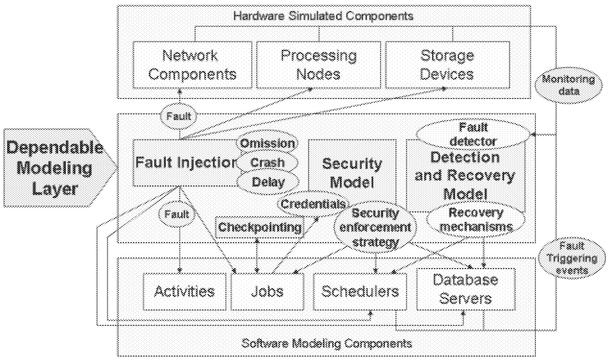


*Figure 3. The dependable simulation model and its components.*

## 3. Fault tolerance model

At hardware level different distributed components can be modeled as failing: the processing unit, the network connectivity as well as the storage devices. At software level we consider the faults occurring in a middleware component (the scheduler behavior could be erroneous, the database server could return wrong values, etc.) or in the higher-level distributed application (for example the jobs could fail to return correct results). For all the modeled components being considered by the simulation model we added specific mechanisms to inject faults. The injection of faults affects primarily the behavior of the components. In this way we are able to model different faults: crash faults, omission faults, time faults, as well as Byzantine faults [1]. In order to cover all possible faults, the model includes the use of any of two mechanisms, as follows.

In the first approach the user input into the simulation model values for the MTTF (mean time to failure) parameter in case of the various components involved in a specific simulation experiment. This parameter represents the basis for simulating the haphazardness stimuli coming from external and internal sources of influence that affect the characteristics of the modeled components and is seen as a probability measure of fault occurrence that is supposedly computed for each component prior to its deployment in the system. For modeling the fault injection mechanisms we use the MTTF together with a mathematical probability distribution (such as binomial, Poisson, Gaussian, standard uniform, etc). At random intervals of time, a component can therefore experience faulty behavior (failures), as well as possible recovery. Regarding the failures, a component can experience complete crash (the component will not recover anymore and will not be accessible by modeled entities), omissions (the network model will deliver only partial messages for example) or delays (the component will experience modeled delays).

The second proposed approach considers a completely random occurrence of fault events, without the user specifying any input value. This is useful in modeling the most disruptive faults, the Byzantine failures that occur arbitrary in the simulated system, such as in the case of transient hardware faults for example: they appear, do the damage, and then apparently just go away, leaving behind no obvious reason for their activation in the first place. For these types of errors the simulation model allows resuming the normal behavior of the affected component, as it is usually enough to allow successful completion of its execution.

The fault injection mechanisms are used together with various fault detection and recovery mechanisms. For that the model is augmented with a monitoring component. The monitoring component is responsible with receiving data of interest (such as fault occurrence triggered events), taking actions to update the state of the distributed system being modeled and possible inform interested components of the events occurrences. This component can track resource faults in the system and generate appropriate controlling actions. For example, based on various fault injection mechanisms, we can trigger the generation of a crash of a processing unit. The trigger translates in the generation of a special simulation event that is received by the monitoring component. Next the monitoring unit removes the processing unit from the modeled system (it will no longer be visible and all its on-going tasks will be forced to stop – the state update action) and inform all interested components of the occurrence of the event. In this approach a scheduler that is interesting in monitoring the state of the processing unit on which it deployed a task for execution would register with the monitoring component and, upon triggering of the crash event, will be informed of the failure so that to take the appropriate correction actions). We actually included in the model (and in the implementation in MONARC) an implementation of a DAG scheduling algorithm that is capable of taking appropriate rescheduling decisions when faults occur. Such a component is useful for example when evaluating different recovery schemes for distributed systems.

In this model an omission fault example is the simulation of a crushed network link, where the monitoring unit is responsible with generating corresponding interrupt events for the still-running tasks that were using that modeled link.

For modeling timing faults the monitoring unit also plays an important role. In the simulation model the boundaries of the action (start of the execution of a task, termination of a task, etc.) are modeled using simulation events. When the monitoring unit receives a timing fault triggering event it will simply modify termination events so that to be triggered at a later time in the future. In this way we simply change the default behavior so that a task will not end when it was supposed to end, but sometimes later in the future. A modeled fault-tolerant software component could then use timing checks (deadlines) to look for deviations from the acceptable behavior. For example, the scheduler also implements a fault-tolerant mechanism, according to which whenever a new job is submitted the scheduler also produces a special simulation event that triggers when the timeout occurs (where by timeout we mean an amount of time dependant on the user specification that will be triggered if the job fails to return results in due time). The scheduler will then be interrupted by one of two actions: either the job finishes or the timeout event occurs (which one happens faster in the simulation timeline). The same mechanisms are implemented by the network simulation model, a job being informed if a transfer failed to finished in a specified amount of time (possible due to network congestion for example) in order to consider taking appropriate measures (such as canceling the transfer for example).

We state that the described extension to MONARC's simulation model is useful for testing both reactive and proactive fault tolerance existing techniques [13]. In case of the reactive fault tolerant systems the fault monitoring systems is informed after the detection of the failure in order to start the corresponding recovery processes. The predictive (proactive) fault tolerant systems predict a fault before its occurrence. Predictive systems are based on the assumption that faults do show some disruptive effect on the performance of the system. In our approach the user can evaluate the performance of the distributed system in the presence of various predictive fault tolerance techniques by augmenting the monitoring component with various prediction algorithms.

In order to allow the evaluation of a wide-range of dependability technologies, the simulation model also includes the mechanisms to allow the modeling of check-pointing or logging of the system's state. These mechanisms are implemented based on the simulation events and the state of the objects used to simulate the modeled components. For that the job interface provides a method that, when called, results in the saving of the serialized objects as well as the state of the simulation on the local disk storage. This is useful in experiments dealing with both static and dynamic check-pointing strategies.

The simulation model also includes the evaluation of various replication and redundancy mechanisms. The replication provides mechanisms for using multiple identical instances of the same system or subsystems and choosing the result based on quorum. The simulation model allows the simulation of DAG distributed activities. This construction is also useful in modeling the replication of the jobs, where the same job would be executed on multiple processing units and another job is used to receive the outputs and select the correct result. Redundancy results were demonstrated by the experiments described in [8]. In the experiments we describe how the simulation model already deals with replicating database storages. We also added replication mechanism to in case of the simulated jobs and scheduler as well.

## 4. Security model

Distributed systems are more vulnerable to security threats than traditional ones due to aspects such as the need for distributed access control, remote access to resources or the wide spread of resources, located in different administrative domains. We present the extension to the original MONARC's model designed to consider the evaluation of security in distributed systems.

In the case of distributed systems the main security threats try to exploit the weaknesses of the protocols and operating systems underneath them, but also the ones exploiting higher levels such as the ones implying attacks over databases, file sharing or multimedia applications, etc. Therefore, a complete security model must consider security aspects ranging from confidentiality of the data, authentication, non-repudiation, data integrity, access control, as well as key management [14].

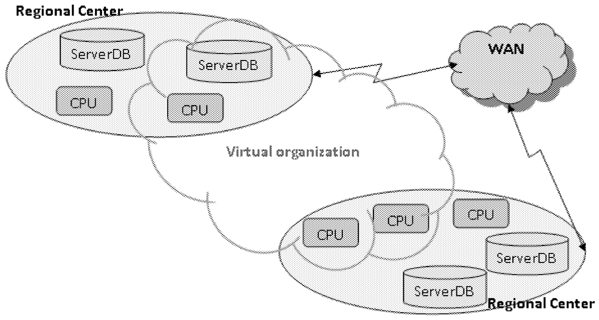


*Fig. 4. The modeling of virtual organizations.*

The starting point in designing the security model was the one available in the real-life middleware Globus Toolkit [15]. The model allows the definition of virtual organizations (VOs), each one can being able to share resources belonging to different regional centers. The model assumes authentication based on X.509 certificates, as in case of Globus. The users submitting jobs must therefore present a valid certificate used to verify their identity within the relation with the different entities throughout the modeled system.

The security model also allows mutual authentication, much like in case of GSI [15], for the initial phase of the communication. The model considers the default SSL mutual authentication protocol. But the security model is extensible, allowing for example also the use of unidirectional authentication, or the protection of messages using a user-defined SSL-like encryption protocol. The access authorization to the resources of the system is also part of the model, being simulated through the use of security policies at the level of components belonging to VOs.

The main mechanisms considered for the security model are: the possibility to create virtual organizations, security policies for VOs, the use of authentication, authorization, the possibility to protect messages using cryptographic protocols, to secure data transfer or to filter traffic. The proposed model therefore describes a wide range of security solutions, such as GSI, PKI, SSL, cryptographic solutions, etc., and is adequate for modeling attach ranging from DoS to detecting attacks using cryptographic messages or authentication and authorization protocols, as well as the modeling of possible reactions to such attacks.

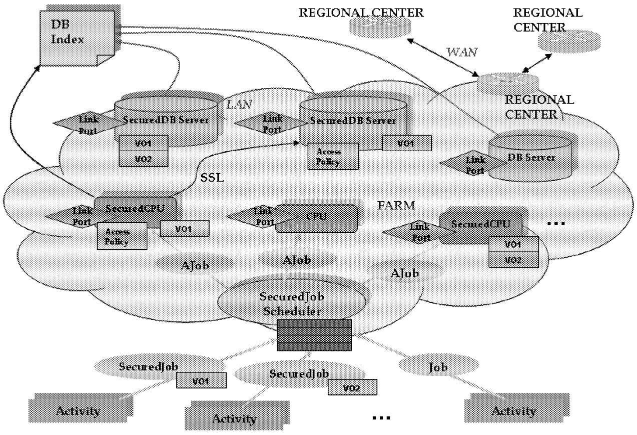


*Fig. 5. The components of the security model.*

In order to implement the model within MONARC all basic components were extended (Figure 5): the processing CPU unit, the job, the database server, the job scheduler – they were all added the mechanisms for authentication, access control and job scheduling according to the restrictions imposed by the VO where the jobs should be executed.

The secured job identify the job submitted for execution within a VO (the default mechanism is based on the use of certificate, but other user-defined mechanisms can also be used). For that we added the possibility to define the VO, as well as added the mechanisms for authentication within the system (and delegation as well).

The secured job scheduler schedules jobs for execution based on access control policies defined using various mechanisms (such as the mapping of certificates versus VOs).

The processing unit was also extended such that to specify the VO to which it belongs. By default for each VO the secured processing unit has an associated security policy and access control method. We actually evaluated in different scenarios access control methods based on the required processing power or memory needed for successful processing. When a job is submitted for execution on a processing unit a series of tests for the determination of identity and authorization, as well as the existence of proper access rights are executed. A regional center can have multiple processing units, shared within one or more virtual organization, as well as processing units that do not belong to any virtual organization (not shared within the cloud).

The database server was also extended to support the sending and receiving of secured messages, as well as various access control policies defined for the access to the data locally stored. In this case the database server maintains a mapping between virtual organizations and access policies. When one tries to access the data on the server a series of tests are being automatically executed, having the role of determining the identity and access roles and, based on the type of operation and data, determine if the user has the right to access the secured data or not. Such a mechanism can be used to model various types of security access solutions, from distributed file systems to user-oriented operating system access control policies.

Also the security model considers the use of cryptographic protocols (such as SSL) to encrypt communication. Such mechanisms are especially useful for ensuring security and confidentiality of the data transferred being model within a simulation scenario. The user has also the possibility to define its own data transfer security protocol, as well as cryptographic algorithms being used.

Finally, in the model we added the possibility of traffic filtering (like in the case of firewalls or routers) within each component. The filtering is based on rules defined statically or dynamically (at simulation time) by the user. To complete the model, we also defined a series of patterns for attacks (such as DoS), useful for the evaluation of various user-defined security mechanisms.

## 5. Results

In a first series of experiments we evaluated the capability of the simulation model to cope with fault tolerance solution in different parts of a scenario. We first evaluated the capability of producing and recovering from failures in the network layer. For these experiments we considered a scenario composed of four regional centers all connected to each others. Within each regional center we considered the existence of a LAN, while between the WANs connecting the regional centers we assumed the existence of two modeled routers.

The network traffic was generated so that the packets would traverse at least one router. In the experiments faults were generated in the LANs, as well as in the routers. We were particularly interested in the correlation between the number of lost packets in the network and the probability coefficients of failures occurrences in the intermediary components.

The retransmission is accomplished by the upper-level protocols – TCP in this case. In the first experiment we simulated the failure of the link port interfaces. When the interface experiences transient faults, after several attempts, the messages are correctly delivered. The results in case of permanent crashes are presented in Figure 6.

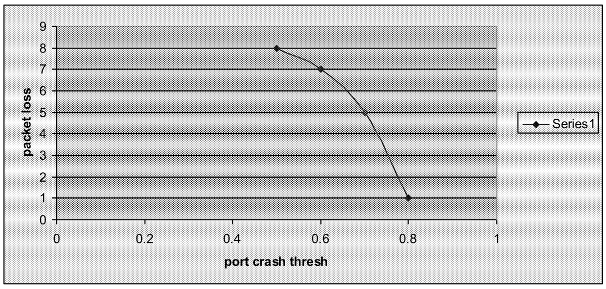


*Fig. 6. The influence of modeled failures over a network transmission in a LAN.*

We next experimented with faults occurring in the routers. The failure of one routers leads to an interruption in the communication. As in the previous experiment, the transient failures only leads to an increase in the time needed to transfer the messages, but eventually the transmission ends correctly. The influence of crash failures is presented in Figure 7.

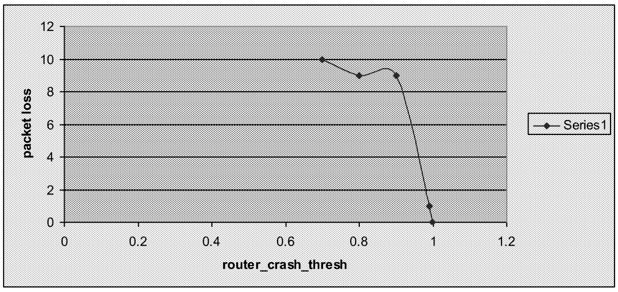


*Fig. 7. The influence of modeled failures in a router over a network transmission.*

A first series of simulation experiments evaluated the fault-tolerant scheduling algorithm for DAGs that is part of the model. The experiments considered the case of several complex DAG dependent tasks that were submitted for execution, and the cases when faults occur or not. We first analyzed the report between the finalized versus submitted tasks for both cases.

We conducted these experiments on the conditions and algorithms proposed in [12]. The results are presented in Figure 8.

In figure 8 the difference between the submitted jobs and the finalized ones represents the number of jobs that were successfully rescheduled (when faults occurred).

An experiment designed to evaluate the security model considered the case of two regional centers (Figure 9) sharing several processing units and a database server within a virtual organization. The purpose of this scenario was to demonstrate the functionality of an access policy within the secured database server and to identify flows within the model.

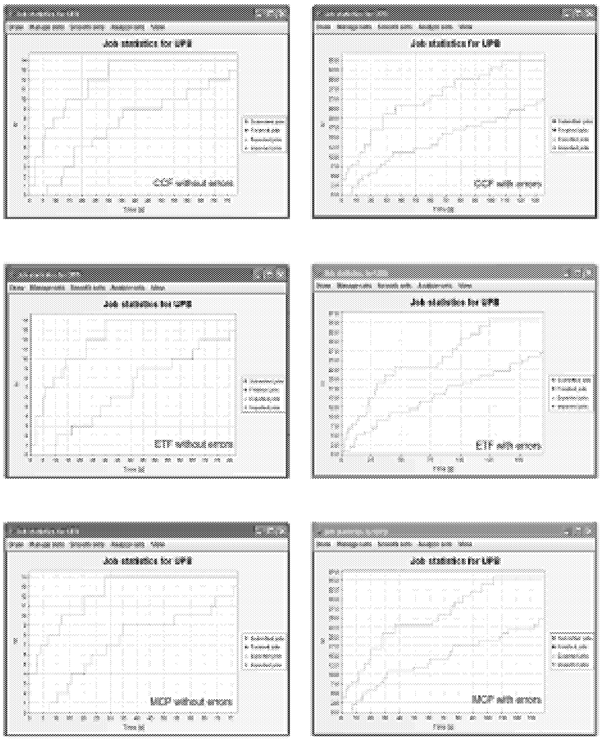


*Fig. 8. Results obtained when evaluating the fault-tolerant scheduling algorithms – the case of running the CCF, ETF and MCP scheduling algorithms [12].*

For this simulation experiment we defined two jobs: one requests the creation of a database and writes data in it and the other one connects to the server and requests the data matching a specific pattern.

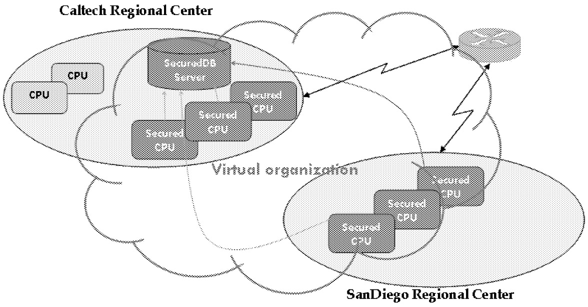


*Fig. 9. The scenario configuration.*

We associated a security policy resembling the UNIX file access policies to the database server belonging to the VO. We next considered that the members of the VO have read and write rights over the database server. Any get operation will be ignored and the operation is considered an implicit attack on the database server.

The experiment consisted in the insertion of many jobs of the types previously presented. The obtained results, presented in Figure 10, demonstrate that, as expected, during an attack the throughput increases, in contrast with the initial conditions of the experiments. Also, the number of received connections increases during an attack. The results demonstrate the validity of the proposed security model, as they are well mapped with the analytical results expected from the experiment. We also conducted a number of other experiments, trying to evaluate the components proposed within the security model, ranging from securing communication to imposing access control at virtual organization level.

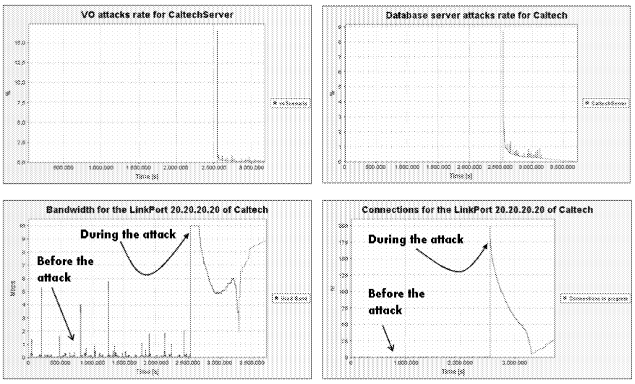


*Fig. 10. Results obtained in case of the security simulation experiment.*

## 6. Conclusions

As society increasingly becomes dependent of distributed systems (Grid, P2P, network-based), it is becoming more and more imperative to engineer solutions to achieve reasonable levels of dependability for such systems. Simulation plays an important part in building and evaluating dependable distributed systems.

In this paper we presented a new simulation model designed to evaluate the dependability in distributed systems. This model extends the MONARC simulation model with new capabilities for capturing reliability, safety, availability, security, and maintainability requirements. The model has been implemented as an extension of the multithreaded, process oriented simulator MONARC, which allows the realistic simulation of a wide-range of distributed system technologies, with respect to their specific components and characteristics.

The extended simulation model includes the necessary components to inject various failure events, and provides the mechanisms to evaluate different strategies for replication, redundancy procedures, and security enforcement mechanisms, as well. The results obtained in simulation experiments presented in this paper probe that the use of discrete-event simulators, such as MONARC, in the design and development of distributed systems is appealing due to their efficiency and scalability.

## 7. References

[1] A. Avizienis, J. C. Laprie, B. Randell, "*Fundamental Concepts of Dependability*", Research Report No 1145, LAAS-CNRS, April 2001.

[2] H. Casanova, A. Legrand, M. Quinson, "*SimGrid: a Generic Framework for Large-Scale Distributed Experimentations*", Proc. of the 10th IEEE International Conference on Computer Modelling and Simulation (UKSIM/EUROSIM'08), 2008.

[3] R. Buyya, M. Murshed, "*GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing*", The Journal of Concurrency and Computation: Practice and Experience (CCPE), Volume 14, 2002.

[4] W. Venters, *et al*, "*Studying the usability of Grids, ethongraphic research of the UK particle physics community*", UK e-Science All Hands Conference, Nottingham, 2007.

[5] K. Ranganathan, I. Foster, "*Decoupling Computation and Data Scheduling in Distributed Data-Intensive Applications*", Int. Symposium of High Performance Distributed Computing, Edinburgh, Scotland, 2002.

[6] S. Naqvi, M. Riguidel, "*Grid Security Services Simulator G3S) – A Simulation Tool for the Design and Analysis of Grid Security Solutions*", Proc. of the First International Conference on e-Science and Grid Computing, Melbourne, Australia, 2005.

[7] C. Dobre, V. Cristea, "*A Simulation Model for Large Scale Distributed Systems*", Proc. of the 4th International Conference on Innovations in Information Technology, Dubai, United Arab Emirates, November 2007.

[8] I. C. Legrand, H. Newman, C. Dobre, C. Stratan, "*MONARC Simulation Framework*", International Workshop on Advanced Computing and Analysis Techniques in Physics Research, Tsukuba, Japan, 2003.

[9] C. Dobre, "*Advanced techniques for modeling and simulation of Grid systems*", PhD Thesis publicly defended at University POLITEHNICA of Bucharest, January 2008.

[10] C. Dobre, F. Pop, V. Cristea, "*A Simulation Framework for Dependable Distributed Systems*", First International Workshop on Simulation and Modelling in Emergent Computational Systems (SMECS-2008), Portland, USA, 2008.

[11] C. Dobre, C. Stratan, V. Cristea, "*Realistic simulation of large scale distributed systems using monitoring*", in Proc. of the 7th International Symposium on Parallel and Distributed Computing (ISPDC 2008), Krakow, Poland, July 2008.

[12] F. Pop, C. Dobre, G. Godza, V. Cristea, "*A Simulation Model for Grid Scheduling Analysis and Optimization*", Parelec , 2006.

[13] N. Naik, "*Simulating Proactive Fault Detection in Distributed Systems*", Proposal Draft for Capstone Project, 2007.

[14] M.S. Perez, B. Xiao, "*Security on grids and distributed systems*", Science Direct, Future Generation Computer System, Universidad Politécnica de Madrid, Hong Kong Polytechnic University, 2007.

[15] The Globus Security Team, „*Globus Toolkit Version 4 Grid Security Infrastructure: A Standards Perspective*", Globus Alliance.

[16] Monarc web page: http://monarc.cacr.caltech.edu

# MONTE CARLO AND LATIN HYPERCUBE METHODS THROUGH A CASE STUDY

Samy Mziou
Al Imam Mohammed bin Saud University,
College of sciences, Department of Mathematics;
P.O. Box. 90950, Riyadh 11623, Saudi Arabia.
Email: mzious@yahoo.fr

## KEYWORDS

Monte Carlo Method, Latin hypercube method, Stochastic differential equations, probability density function, Soluble solids content.

## ABSTRACT

Monte Carlo and Latin Hypercube methods are two famous computational algorithms for simulating some problems with significant stochasticity. The main difference between them is the Latin Hypercube method takes into account the full coverage sampling. In this work, both methods are compared via a real example, the study of one apple quality attribute called the soluble solids content.

## INTRODUCTION

Up to date a deterministic approach is not realistic for simulating some problems with significant uncertainties or randomness. Stochastic approaches must be considered. Moreover, the booming of super computers with huge memory and multiprocessors has allowed to computational algorithms to become the only alternative in simulation of physical phenomena reducing in that way the cost of a real experiment.

Monte Carlo Method (MCM) (Hammersley and Handscomb 1964, Kalos and Whitlok 1986, Rubinstein 1981, Soból 1981) is a well known computational method for generating random numbers used for simulating random phenomena. This method is used to generate uniform random numbers using for instance the "roulette" or some mathematical formulae such the famous congruence methods. To generate non uniform random numbers, different methods exist such as Inverse Transform method, Acceptance Rejection method (whose the famous Importance sampling method), stratified sampling method, Convolution method, Composition method, etc... .

Although this method is easy to implement the MCM method has two major drawbacks. First, it needs a paramount number of random samples to recreate a certain distribution leading to a growth of time consuming and memory, and secondly, it doesn't guarantee good random numbers distribution in which a complete sampling is often needed.

The Latin hypercube method (LHM) (McKay et al. 1979) is another technique used to generate random numbers or more precisely an another way to sample like Stratified sampling Method (SSM). This method provides an efficient way of sampling by a full coverage of the range of each variable by maximally stratifying the input distribution. Latin Hypercube Sampling, developed by McKay, Conover, and Beckman (1979), selects $n$ different values from each of $k$ variables $X_1$, $X_2$, ..., $X_k$ in the following manner. The range of each variable is divided into $n$ nonoverlapping intervals on the basis of equal probability. One value from each interval is selected at random with respect to the probability density in the interval. The $n$ values obtained for $X_1$ are paired in a random manner (equally likely combinations) with the $n$ values of $X_2$. These $n$ pairs are in turn, combined in a random manner with the $n$ values of $X_3$ to form a triplets, and so on, until $n$ $k$-uplets are formed. Because of its lesser time consuming cost and its full coverage sampling, the LHM has generated a strong interest. We can cite for instance (Hossein et al. 2006, Minasny and McBratney 2006, Pebesma and Heuvelink 1999, Zhang and Pinder 2003).

In this paper, both methods will be considered and compared by studying a real problem: the apple storage. Indeed, recently (Mziou 2009) the basics dynamics of fruit characteristics have been modeled by a stochastic approach. The time evolution of apple quality attributes was represented by means of a system of differential equations in which the initial conditions and model parameters are both random. The case study considered in this paper, is one apple quality attribute called the soluble solids content. For this apple characteristic, the system of differential equations is linear and the state variable and the parameters are represented as random variables with their statistical properties (mean

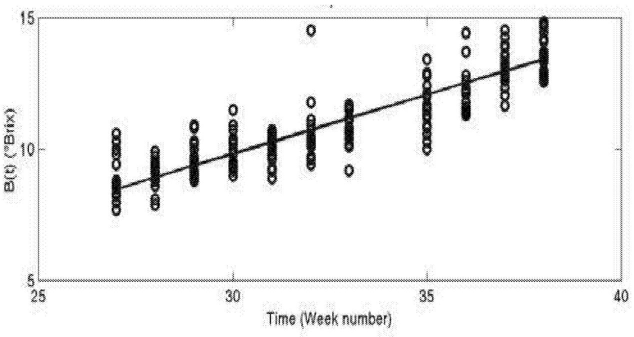


Figure 1: Evolution of Soluble Solids content during apple maturation and cold storage.

values, variances, covariances, joint probability density function) known at the initial time. The Monte Carlo and Latin Hypercube Methods were developed to obtain a numerical expression of the dynamic behavior of this statistical quantity and particularly to follow the time evolution of joint probability density function which represents one of the best mean to characterize random phenomena linked with fruit quality attributes.

## PROBLEM STATEMENT

In order to follow the time evolution of apple maturity and determine the optimal picking date at the harvest, a major fruit quality attribute is used to quantify apple sweetness: the soluble solids content $\boldsymbol{B}(t)$ which characterizes the sugar rate in apple (component of taste).

A Jonagold apple quality attribute was followed during the 2001 harvest season. Apples were carefully hand harvested from a commercial orchard (Schotsmans apple grower, Rummen, Belgium). During three months before harvest, twenty apples were weekly collected. After picking, the fruits were transported to the laboratory where the soluble solid content was measured with a digital refractometer (Stable Micro Systems Texture Analyser, Surrey, England, UK). The unit chosen for this measurement is $^{o}Brix$ which correspond to the dissolved sugar-to-water mass ratio of a liquid. For example a $15^{o}Brix$ solution is 15% (w/w), with 15 grams of sugar per 100 grams of solution. The measurement data are shown in Figure 1.

Figure 1 shows the time evolution during apple maturation of the soluble solids content obtained from experiments and the parametric model used to represent their dynamic behavior. The parametric model chosen is usually a linear one.

The soluble solids content $\boldsymbol{B}(t)$ which depends on time, satisfies a system of linear differential equations (1-4) with model parameters and initial conditions at time $t_0$. This system (Mziou 2009) is given by

$$\frac{d\boldsymbol{B}(t)}{dt} = k \quad ; \quad t \in T \quad , \tag{1}$$

$$\frac{dk}{dt} = 0 \quad , \tag{2}$$

$$\boldsymbol{B}(t_0) = \boldsymbol{B_0} \quad , \tag{3}$$

$$k(t_0) = k_0 = k \quad , \tag{4}$$

where $k$ is the model parameter, $\boldsymbol{B_0}$ and $k_0$ are the initial conditions, and $T = [t_0, t_{max}] \subset \mathbb{R}_+$ is the time interval of analysis with $t_0$ (initial time) and $t_{max}$ (final time) are such that $t_0 < t_{max}$.

The measurement data shown in Figure 1 indicate that the variability in the dynamic behavior of the soluble solids content is determined by the initial condition and parameter (slope) which are both random. Hence, for each fruit quality attribute, the stochastic parametric model needs to take into account this variability which will be introduced via the initial conditions and the parameters. Consequently, in equations (1-4), the initial condition $\big(\boldsymbol{B_0}\big)$ of the state variable $\big(\boldsymbol{B}(t)\big)$ and the model parameter $\big(k\big)$ will be quantified as random variables with their joint density probability functions, their mean values and their covariance matrices. However, according to equation (2), we assume that the model parameter $\big($i.e $k\big)$ do not change in time.

## RESULTS AND DISCUSSION

The initial condition $B_0$ and the parameter $k$ are assumed Gaussian random variables with mean values $\mu_{B_0} = 8.47^o$Brix, $\mu_k = 0.45^o$/week and respective standard deviation $\sigma_{B_0} = 0.36^o$Brix, $\sigma_k = 0.05^o$Brix/week (Mziou 2009). From measurement data it was estimated that at the initial time $t_0$, the initial condition $B_0$ and the parameter $k$ are correlated with a given covariance equal to $C_{k,B_0} = -0.014^o\text{Brix}^2$/week (Mziou 2009). The joint probability density function is given by

$$p_{Y_0}(\boldsymbol{\alpha}) = \frac{1}{2\pi\,(det\,\mathbf{C}_{Y_0})^{\frac{1}{2}}} e^{-\frac{1}{2}[(\boldsymbol{\alpha}-\boldsymbol{\mu}_{Y_0})^T \mathbf{C}_{Y_0}^{-1}(\boldsymbol{\alpha}-\boldsymbol{\mu}_{Y_0})]}, \tag{5}$$

where $\boldsymbol{Y_0} = [B_0 \quad k]^T$, $\boldsymbol{\mu}_{Y_0} = [\mu_{B_0} \quad \mu_k]^T$ and $\mathbf{C}_{Y_0} = \begin{bmatrix} C_{B_0} & C_{B_0,k} \\ C_{B_0,k} & C_k \end{bmatrix}$.

Figure 2 shows the time evolution of the solution $B(t)$ on the interval $T = [0 : 20]$ weeks when the initial condition $B_0$ and the parameter $k$ are both Gaussian random variables. For each of these random variables,

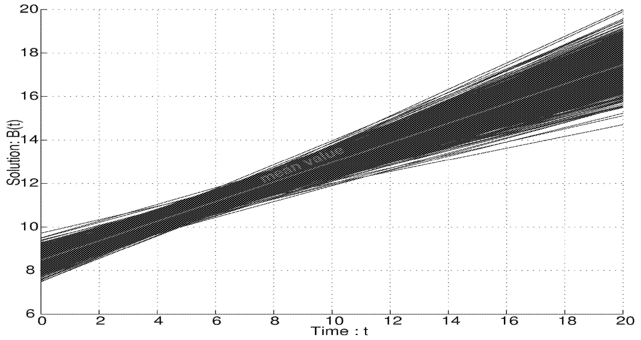


Figure 2: Solution $B(t)$ when both initial condition and parameter are Gaussian.

the number $n_{MC}$ of generated random numbers is equal to 50000. The grey line represents the time evolution of the mean value of the stochastic process $B(t)$ (Friedman 1975, Guikhman and Skorokhod 1979, Melsa and Sage 1973, Papoulis 1991).

The variance propagation algorithm (Melsa and Sage 1973, Nicolaï and Baerdemaeker 1999, Scheerlinck et al. 2001) applied to the soluble solid content model (Eqs. 1-4) gives in this case the exact dynamic behavior of the mean values, the variances and the covariance between the state variable $B(t)$ and the model parameter $k$. These statistical quantities satisfy the following equations

$$\frac{d\mu_B(t)}{dt} = \mu_k \ ; \quad \frac{d\mu_k(t)}{dt} = 0 \ ; \tag{6}$$

$$\frac{dC_k(t)}{dt} = 0 \, ; \ \frac{dC_{Bk}(t)}{dt} = C_k(t) \, ; \ \frac{dC_B(t)}{dt} = 2\, C_{Bk}(t) \, ; \tag{7}$$

with initial conditions

$$\mu_B(t_0) = \mu_{B_0} \quad \text{and} \quad \mu_k(t_0) = \mu_k \quad , \tag{8}$$

$$C_B(t_0) = C_{B_0} \, ; \, C_{B,k}(t_0) = C_{B_0,k} \text{ and } C_k(t_0) = C_k \ , \tag{9}$$

and whose trivial solutions are given by

$$\mu_B(t) = \mu_{B_0} + (t - t_0)\,\mu_k \quad , \tag{10}$$

$$\mu_k(t) = \mu_k = \text{constant} \quad , \tag{11}$$

$$C_k(t) = C_k = \sigma_k^2 = \text{constant} \quad , \tag{12}$$

$$C_{B,k}(t) = C_{B_0,k} + (t - t_0)\, C_k \quad , \tag{13}$$

$$C_B(t) = C_{B_0} + 2\,(t - t_0)\, C_{B_0,k} + (t - t_0)^2\, C_k \quad . \tag{14}$$

Eqs. (10-14) show that the mean value $\mu_B(t)$ and covariance $C_{Bk}(t)$ are linear functions whereas the variance $C_B(t)$ is quadratic one.

Consequently, the stochastic process $\boldsymbol{Y}(t) = [B(t)\ k]^T$ has a mean value and a covariance given by

$$\boldsymbol{\mu}_Y(t) = \begin{bmatrix} \mu_B(t) \\ \mu_k \end{bmatrix} \ , \tag{15}$$

$$\mathbf{C}_Y(t) = \begin{bmatrix} C_B(t) & C_{B,k}(t) \\ C_{B,k}(t) & C_k \end{bmatrix} \ , \tag{16}$$

and a joint probability density function, which is gaussian, given by

$$p_{Y(t)}(\boldsymbol{\alpha}, t) = \frac{e^{-\frac{1}{2}\left[ \left(\boldsymbol{\alpha} - \boldsymbol{\mu}_Y(t)\right)^T \ \left(\mathbf{C}_Y(t)\right)^{-1} \ \left(\boldsymbol{\alpha} - \boldsymbol{\mu}_Y(t)\right) \right]}}{(2\pi)^{\frac{n}{2}} \left[ \det\left( \mathbf{C}_Y(t) \right) \right]^{\frac{1}{2}}} \ . \tag{17}$$

The characteristic functions method (Melsa and Sage 1973) is used to derive a solution for the distribution of the solution $B(t)$ at each time instance. As $\boldsymbol{Y_0}$ is Gaussian, its characteristic function is given by

$$\Gamma_{Y_0}(\mathbf{u}) = e^{<\mathbf{u}\, ,\, i\ \boldsymbol{\mu}_{Y_0} - \frac{1}{2}\, \mathbf{C}_{Y_0}\, \mathbf{u} >_{\mathbb{R}^n}} \quad , \tag{18}$$

where $< \mathbf{u}\, ,\, \mathbf{v} >_{\mathbb{R}^n} = \mathbf{u}^T\, \mathbf{v}$ denotes the scalar product on $\mathbb{R}^n$.

The characteristic function of $\boldsymbol{Y}(t)$ (see Melsa and Sage 1973) is given by

$$\Gamma_{Y(t)}(\mathbf{u}, t) = \Gamma_{Y_0}(\mathbf{A}_t^T\, \mathbf{u}) \quad , \tag{19}$$

in which $\mathbf{A}_t = \begin{bmatrix} 1 & (t - t_0) \\ 0 & 1 \end{bmatrix}$ is defined from the following relation $\boldsymbol{Y}(t) = \mathbf{A}_t\, \boldsymbol{Y_0}$.

By taking $u_k = 0$ in $\mathbf{u} = [u_B\ \ u_k]^T$ and by introducing this value of $\mathbf{u}$ into (Eq. 19), we obtain that the characteristic function of the solution $B(t)$ is given by

$$\Gamma_{B(t)}(u_B, t) = e^{<u_B\, ,\, i\ \mu_B(t) - \frac{1}{2}\, C_B(t)\, u_B >_{\mathbb{R}^n}} \quad . \tag{20}$$

Consequently, the solution $B(t)$ is Gaussian with mean value $\mu_B(t)$ and variance $C_B(t)$ and its probability density function is given by

$$p_{B(t)}(\alpha, t) = \frac{1}{\sqrt{2\pi\ C_B(t)}}\, e^{-\frac{1}{2}\left[ \frac{(\alpha - \mu_B(t))^2}{C_B(t)} \right]} \quad . \tag{21}$$

An analytical expression such as (Eq. 21) is rarely obtained and numerical approaches to estimate the joint probability density function are usually needed for more complex models. Anyway, an analytical expression is useful to test numerical approaches. In this work two numerical techniques, the Monte Carlo and Latin Hypercube methods are considered and compared (Figures 3 and 4) in order to compute the joint probability density function $p_{B(t)}(\boldsymbol{\alpha}, t)$ between the parameter $k$ and the solution $B(t)$, at six different times $t = 0$ week, t=4 weeks, $t = 8$ weeks, $t = 12$ weeks, $t = 16$ weeks and $t = 20$ weeks. For both methods, the number of random samples selected are $n = 10000$ and $n = 100000$. MATLAB 6.1 (The Mathworks, Natick, USA) was used to perform all computations and simulations. In addition, we have assumed that at the initial time $t_0$, the random variables $B_0$ and $k$ are correlated Gaussian random variables with a given coefficient of correlation, which is obtained by experimental and statistical means. Moreover, an adaptable meshing is chosen which consist in meshing at each time instance each joint probability density function separately from each other. Indeed, at each time of interest, intervals for $k$ and for $B(t)$ are given and are divided into 20 subintervals which constitute the plane meshing. These intervals are obtained by considering a 99.7% confidence interval ($\pm 3\sigma$) for both $k$ and $B(t)$ around their mean values at each time instance.

Figure (3) shows the joint probability density functions obtained analytically (above subfigures) and by the Monte Carlo method (below subfigures). We have respectively considered 10000 (left below subfigure) and 100000 (right below subfigure) generated random numbers. We can see that the joint probability density functions obtained by this method are not good at all excepted two of them.

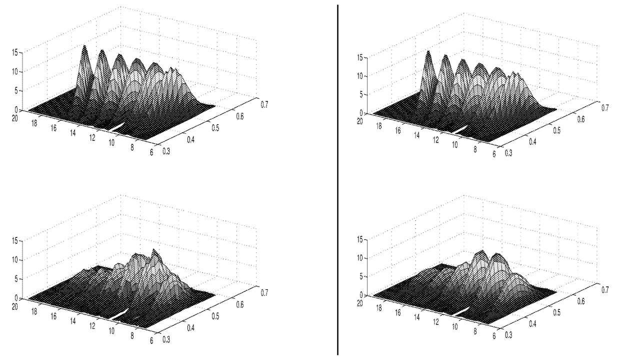


Figure 3: Monte Carlo Method: 10000 samples (left figure) and 100000 samples (right figure)

Figure (4) shows the joint probability density functions obtained analytically (above subfigures) and by the Latin Hypercube method (below subfigures). We have respectively considered 10000 (left below subfigure) and 100000 (right below subfigure) generated random numbers. We can see that the joint probability density functions obtained by this method are very good.

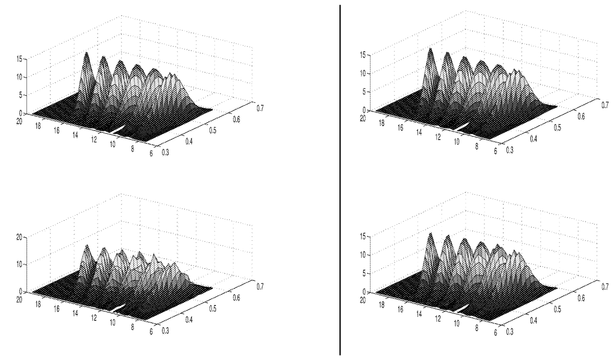


Figure 4: Latin joint probability density function visualization $n_{10000}$ (left figure) and $n_{100000}$ (right figure)

Figure (5) shows the Mean Square Error with 100000 generated random numbers, by using respectively the Latin Hypercube (full line) and Monte Carlo (dashed line) methods. We can observe that the Latin Hypercube Method is more accurate than the Monte Carlo method.

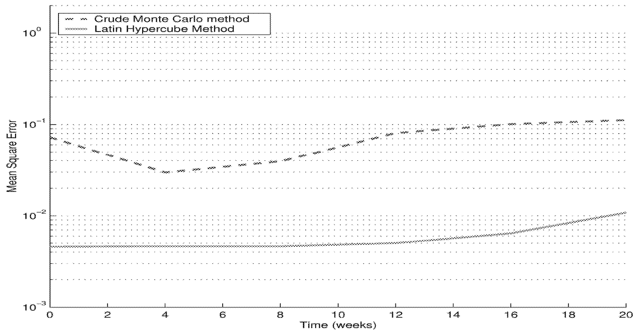


Figure 5: Root Mean square Error by using Monte Carlo and Latin Hypercube methods with 100000 samples

## CONCLUSION

In this work two famous computational techniques, the Monte Carlo and Latin Hypercube methods, are used and compared to estimate the joint probability density functions of the soluble solids content at different time instance. We can observe that for this study case, the Latin Hypercube method is more efficient and more accurate than the Monte Carlo method due essentially that it takes into account the full coverage sampling (one sample is selected in each sampling interval) leading to a better random numbers generation. Furthermore, we can notice that when the time increases, the joint probability density function rotates counterclockwise in the Y-axis direction. All the state variables considered in this example are gaussian. It will be interesting to extend it to other kind of probabilities. Moreover, we have assumed that the model parameter do not change in time.

## REFERENCES

Friedman A., 1975. *Stochastic differential equations and applications.* Vol 1. Academic Press, New York.

Guikhman L. and Skorokhod A., 1979. *The theory of stochastic processes.* Springer Verlag, Berlin.

Hammersley J. and Handscomb D., 1964. *Monte Carlo Methods.* Chapman and Hall.

Hossein F.; Emmanouil N.; and Bagtzoglou A., 2006. *On Latin Hypercube sampling for efficient uncertainty estimation of satellite rainfall observations in flood prediction. Computers & Geosciences*, 32, 776–792.

Kalos M. and Whitlok P., 1986. *Monte Carlo Methods, Vol 1: Basics.* John Wiley and Sons.

McKay M.; Conover W.; and Beckman R., 1979. *A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics*, 1, no. 21 (2), 239–245.

Melsa J. and Sage A., 1973. *An Introduction to Probability and Stochastic Processes.* Prentice-Hall, Englewood Cliffs, New Jersey.

Minasny B. and McBratney A., 2006. *A conditioned Latin Hypercube method for sampling in the presence of ancillary information. Computers & Geosciences*, 32, 1378–1388.

Mziou S., 2009. *Study and modelling of two apple quality attributes: the soluble solids content and the firmness. Mathematical & Computer Modelling of Dynamical Systems*, 15, no. 4, 317–336.

Nicolaï B. and Baerdemaeker J.D., 1999. *A variance propagation algorithm for the computation of heat conduction under stochastic conditions. International Journal of Heat and Mass Transfer*, 42, 1513–1520.

Papoulis A., 1991. *Probability, Random Variables and Stochastic Processes.* Mac Graw-Hill, 3rd edition.

Pebesma E. and Heuvelink G., 1999. *Latin hypercube sampling of Gaussian random fields. Technometrics*, 41, 303–312.

Rubinstein R., 1981. *Simulation and the Monte carlo method.* John Wiley and sons, Inc.

Scheerlinck N.; Verboren P.; Stigter J.; Baerdemaeker J.D.; Impe J.V.; and Nicolaï B., 2001. *A variance propagation algorithm for stochastic heat and mass transfer problems in food processes. International Journal for Numerical methods in Engineering*, 51, 961–983.

Soból I., 1981. *A Primer for the Monte Carlo Method.* CRC Press, Inc., Boca Raton, Florida, 107 pages.

Zhang Y. and Pinder G., 2003. *Latin-hypercube sample-selection strategies for correlated hydraulic-conductivity fields.* Tech. rep., Water Resources Research.

## BIOGRAPHY

**SAMY MZIOU** is assistant professor at Al Imam Bin Saud University (Riyadh, Saudi Arabia) in the department of mathematics. He received his PhD from the CNAM-Paris while he was working at ONERA (French Aerospace Lab, France) in the area of structural mechanics and vibration. He also had a work experience at CEA Saclay (Paris, France) in electromagnetism area. He taught two years in the department of mathematics at PARIS 8-University (France). He worked one year as research associate at Leuven catholic university (Leuven, Belgium) in the laboratory of Postharvest technology. His research interests are applied mathematics, scientific computation and numerical analysis.

# SIMULATION TOOLS

# EVALUATION OF FOUR ARTIFICIAL LIGHTING SIMULATION TOOLS WITH VIRTUAL BUILDING REFERENCE

Shariful H. Shikder, Andrew D. Price, Monjur M. Mourshed
Department of Civil and Building Engineering
Loughborough University
Loughborough, United Kingdom
E-mail: s.h.shikder@lboro.ac.uk

**KEYWORDS**
Simulation Tools, Artificial Lighting, Evaluation, Validation.

## ABSTRACT

There are number of software environments available to conduct lighting simulation to support researchers and designers. This paper compares and analyses performance of four lighting simulation programs, they are, AGI32, DIALux, RADIANCE and RELux. Along with the evaluation of modelling ability and output features, two significant aspects of lighting calculation focussed in this study are accuracy in calculating illumination level and luminaire number for interior spaces. Illumination level calculation is validated by Commission International de Eclairage (CIE) recommended methodology known as CIE test cases. Another virtual space is used to evaluate luminaire number calculation capability. Modelling and simulation were conducted by all four packages. Validation was done by comparing simulation results with analytical and manual method calculation.

## INTRODUCTION

Lighting is a significant built environment design criteria, and architects/designers always keen to expend efforts to deliver optimal lighting solution. Along with the day-lighting, artificial or electric lighting is now obligatory to meet building lighting requirement. Computer simulation packages exist to support lighting designers through facilitating prediction and evaluation ability of design decisions. A number of lighting simulation packages exists to support designers and their capabilities vary in modelling, calculation and output features.

A comprehensive review of few earlier validation studies can be found in Roy (2000). Beside academic researchers, assessment of simulation accuracy also found form individual organizations engaged with tool development (DIALux, INTEGRA). Comparatively more validation studies are focused on day lighting as variability of natural light intensity made it more complex to model appropriate daylight environment (Ashmore and Richens 2001). However considerable interests also exist in modelling and simulating artificial lighting and it can be suggested that simulation tools play vital role in designing artificial lighting environment. Li and Tsang (2004) studied the agreement of lighting simulation output with daylight and artificial lighting with RADIANCE and achieved satisfactory results. They also concluded that computer simulation can play important role in designing energy efficient lighting design. Evangelos and David (2005) reviewed four lighting simulation tools includes Desktop RADIANCE, Rayfront, RELux 2004 Vision and Lightscape based on measured daylight data, user interface and help manuals. A systematic study under International Energy Agency (IEA) Task 31 (IEA 2005) evaluated two lighting computer programs (RELux Professional and Lightscape) and introduced the application of the CIE test cases methodology. This study included 32 test scenarios to evaluate accuracy of lighting calculation programs for artificial and day lighting (Maamari et al. 2006). The study is based on theoretical principles and reference data, and established the test scenarios as a strong methodology to evaluate lighting computer programs. Most of the studies primarily focused on assessing illumination level (Li and Tsang 2004; Evangelos and David 2005; Maamari 2006), few studies also attempted to evaluate luminance environment (Roy 2000) with measurement. Apart from the simulation tools mentioned above newer tools are also developed and existing tools are updated with the advancement of software and hardware technology. A list of existing lighting simulation programs with brief description can be found in Department of Energy - Building Technologies Program (DOE 2008) website. This paper studies and compares four contemporary lighting simulation programs. Evaluation criteria included the accuracy of calculation output (illumination level and luminaire number); ability to model light source and scene geometry; and flexibility in data manipulation.

## SELECTION OF SIMULATION PACKAGES

For this study primary focus in considering simulation programs were ability to model and calculate artificial lighting. Four simulation tools were selected based on wide use, acceptability, availability and previous references. Selected lighting simulation packages for this study are:

- AGI32 v2.04
- DIALux 4.6
- Ecotect 5.5 + Radiance
- Relux Professional 2007

AGI32 is a lighting simulation tool for daylight and artificial lighting, developed and distributed by Lighting Analyst Inc. This program has integrated ‘raytrace’ and ‘radiosity’ based calculation engine to produce lighting calculations and photorealistic images. Ecotect is a building simulation program introduced by Square One Research. The program is a building performance analysis tool to use at the earlier stage of design. The program is capable of using RADIANCE for lighting calculations and provides illuminance and luminance values over a customized analysis grid, or image output. Combination of Ecotect and RADIANCE is more used in academic research rather than professional lighting design. Radiance is a ray-traced based lighting simulation program and uses backward raytracing process for simulation, developed by Lawrence Berkeley Laboratory (Larson 1998). Developed by DIAL GmbH DIALux is a widely used commercial package in lighting design, which is available for free through lighting manufacturers' websites. This program is customized for interior and road lighting. Relux Professional is developed by Informatik AG, can be used both for interior and exterior. This program uses average indirect fraction methodology for calculation and can produce photorealistic image output by RADIANCE through Relux Vision interface.

## COMPARISON OF SIMULATION TOOLS

### Scene Modeling

All packages used in this research are capable of modeling 3D environment. RADIANCE does not provide any 3D modelling interface. Ecotect is capable modelling of 3D geometry and compatible to export to RADIANCE format precisely. DIALux, Relux and AGI32 provide their own library for preset 3d objects. In some cases preset object libraries are an advantage; on the other hand they can limit number of polygons and detail definitions.

Table 1: Acceptability of standard 3d formats.

|  | Ecotect | DIALux | Relux | AGI32 |
|---|---|---|---|---|
| Dxf | √ | -[1] | √ | √ |
| Dwg |  | -[1] |  | √ |
| 3ds | √ | √ | √ |  |
| VRML Object | √ |  | √ |  |

[1] Imports as lines only, no 3d surface or object.

One of the important aspects of 3D modelling is importing ability of standard 3D formats, because there is a considerable interest found among the users to use models built for other purposes. In some cases complex models are desired to be modelled by standard CAD packages (Roy 2000). Differences are found in importing and handling ability of these formats among the packages. A comparison is shown in Table 1 of standard 3D model importability among the packages. Images demonstrated in Figure 5 are rendered by importing same models in all four packages developed in a standard CAD package.

### Luminaire Modeling

Defining the luminaire shape and light source appropriately is the key to get accurate output in lighting simulation. Along with the capability of accurately reading photometry files, it is desired to get enough opportunities to modify parameters separately to model the desired luminaire, such as, lamp colour, lamp loss factor or correction factor etc. All four programs evaluated in this study well accept IESNA photometric format. AGI32, DIALux and Relux can take luminaire definition in greater detail by online or downloadable catalogues from manufacturers.

In case of Ecotect + RADIANCE the IESNA photometric profile file needs to be converted by 'ies2rad.exe' program to be readable by RADIANCE. Ecotect v5.5 itself can import photometry files but unable to import intensity distribution of multiple horizontal axis. Also it fails to import luminous flux and dimensional form from the photometry. These inabilities can be resolved by converting IESNA photometry by 'ies2rad.exe' program and using it from external RADIANCE readable light definition file. DIALux, Relux, AGI32 also afford opportunity to modify the parameters of luminaire definition like, luminous flux output, lamp colour, dimension, mounting height etc. The output multiplier can be modified by 'correction factor' in DIALux and 'Lamp Loss Factor' parameters in AGI32.

### Output Features

Comparisons of few output features are given in Table 2 desired by a standard lighting simulation package, which helps in lighting design. These selections of features are referred by Ashmore and Richens (2001). Two significant output features are generation of photorealistic rendered image and data visualization.

Table 2: Comparison of output features of different simulation packages.

| **Output features** | **Ecotect + RAD** | **DIALux** | **Relux** | **AGI32** |
|---|---|---|---|---|
| View of working plane with iso-illuminance contours | √ | √ | √ | √ |
| View iso-illuminance contours in scene | √ | √ | √ | √ |
| False colour in camera view | √ | √ | √ | √ |
| Illuminance at a reference point | √ | √ | √ | √ |
| Photorealistic renderings | √ | √ | √ | √ |
| Luminance at a reference point | √ | √ | √ | √ |
| Virtual Reality Markup Language | √ |  | √ |  |
| Walkthrough animation (rendered) |  |  |  | √ |
| Luminance and illuminane level with pointer scene or camera | √ |  |  |  |
| Simulation data manipulation in scene | √ |  |  |  |
| Simulation data export | √ |  |  |  |

The selected packages use radiosity and raytrace based rendering technology to generate simulated images, where appropriate use of parameters can produce photo-realistic images. A comparison of rendered images by four simulation programs is presented in Figure 5 with a test case of single bed patient room.

Besides generating camera views visualization of data is mentioned as one of the important criteria of simulation tools. Visualization of data is possible over reference grid points for all the packages with graphically rendered iso-illuminance contours. In case of AGi32, DIALux and RELux this data is not exportable to other formats. Ecotect is capable to save analysis grid data in delimited text format, which provides further opportunity in data handling through external applications. Along with this Ecotect also provides opportunity to save multiple data in one analysis grid, which provides improved manipulation ability of simulation data.

**Calculation of Illumination Level**

Illumination level calculation is one of the primary objectives of lighting simulation programs and is integrated with lighting design. In this study validation of illuminance value calculation by four selected packages are conducted by applying two CIE test case scenarios. Detail methodologies are described in the following sections.

*Application of the CIE Test Case*
Commission Internatiole de Eclairage (CIE) established an evaluation procedure of the output performance of lighting simulation packages published as IEA SHC Task 31 (Maamari 2005). The validation approach is based on testing different aspects of lighting simulation by individual test scenarios. The approach includes validation procedure for both artificial and day-lighting and based on theoretical principals where comparison is done with analytically calculated reference data to avoid uncertainties (Maamari et al. 2006). In this study conducted two experiments have been conducted with the CIE test case scenarios, they are, artificial direct lighting calculation for point light and area light source.

*Simulation Parameters*
Following simulation parameters were used for this test, some of the parameters stated here do not affect illuminance level calculation but radiosity/raytrace based image generation.
*Ecotect + Radiance:*
RADIANCE export parameters:
Model detail: High
Light variability: High
Image quality: High
Indirect reflections: 4
*AGI32:*
Mesh level: 1.1 - 3.2 (depending on the size of the plane)
Calculation mode: full
Radiosity Convergence: Maximum Steps: 1000
Stopping Criterion (Convergence): 0.01
Display Interval: 10
Electric and Day lighting for All other surfaces
Maximum Subdivision Level: 5 (level 3 is adequate in most cases)
Minimum Element Area (Sq.M.): 0.0465
Element Luminance Threshold: 1.5
*DIALux:*
Calculation options: very accurate
Calculation method: standard
POV Ray settings:
Smooth edges: On
Indirect calculation: High
Radiosity settings: Count 70; error bound 1.800; pretrace_start 0.800; pretrace_end 0.040; gray_threshold 0.200
*Relux:*
Calculation parameters:
Precision: High indirect fraction
Raster: 0.7
Active Dynamic Raster: on, fine

*Artificial direct lighting – point light source*
Scenario used for this test is as same as described in CIE test case conducted by Maamari (2006) and illustrated in Figure 1. The virtual space is a 4m x 4m in dimension with height of 3m. A point light source is located at the centre at 3m high and analytical reference data were calculated in horizontal surface at floor level.

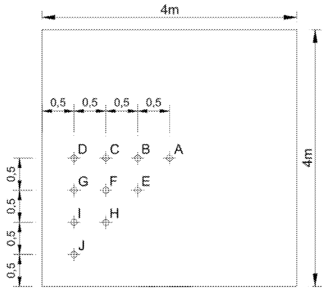


Figure 1: CIE test case suggested reference points' position for point light source as described by IEA (2005).

Analytical reference data are calculated by the equation:

$$E = I\frac{\cos\theta}{d^2}$$

Here, *E* is the horizontal illuminance (lux), *I* is the intensity (candela) of the point light source in the direction of the reference point; *θ* is the incident angle of the light arriving to the reference point from the light source (radians); *d* is the distance between light source and the reference point. Simulation results are presented in Figure 2, which demonstrates very identical results among four simulation programs with analytical reference data and maximum deviation is found within 0.46%.

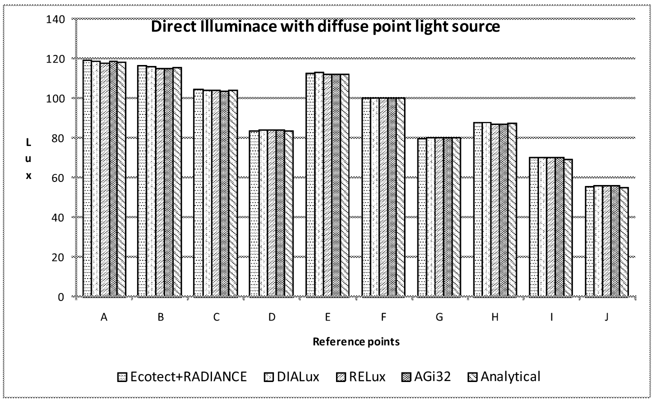


Figure 2: Comparison of illuminance level simulation with analytical results by point light source.

*Artificial direct lighting – area light sources*
This test evaluates capability of the lighting program to calculate illuminance level from an area light source. Test scenario is a square space (4mx4mx3m), where analytical reference data is calculated horizontally as explained in

Figure 3. Light source is a 1m x 1m in size with uniform intensity distribution.

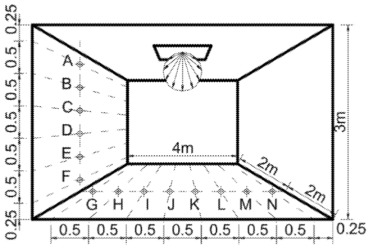
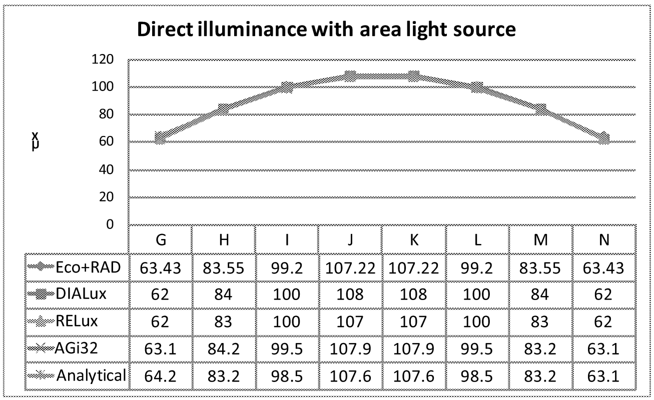


Figure 3: Model description of CIE test case for area light source as described by IEA (2005).

Reference values are calculated by following equation:

$$E_1 = M_2 \times F_{12}$$

Here, $E_1$ is the direct illuminance at any reference point received from the area light source; $M_2$ is the luminous existence or emittence of the area light source (lumen/meter$^2$); $F_{12}$ is the configuration factor between the receiving area and the light source. Details of the calculation procedure can be found in Maamari (2006). Figure 4 represents the result of this test and demonstrates that simulation programs are well equivalent with the analytically derived reference data.



| | G | H | I | J | K | L | M | N |
|---|---|---|---|---|---|---|---|---|
| Eco+RAD | 63.43 | 83.55 | 99.2 | 107.22 | 107.22 | 99.2 | 83.55 | 63.43 |
| DIALux | 62 | 84 | 100 | 108 | 108 | 100 | 84 | 62 |
| RELux | 62 | 83 | 100 | 107 | 107 | 100 | 83 | 62 |
| AGi32 | 63.1 | 84.2 | 99.5 | 107.9 | 107.9 | 99.5 | 83.2 | 63.1 |
| Analytical | 64.2 | 83.2 | 98.5 | 107.6 | 107.6 | 98.5 | 83.2 | 63.1 |

Figure 4: Comparison of illuminance level simulation with analytical calculation by area light source.

### Calculation of Luminaire Number

Calculation of luminaire number takes place during lighting design of interior spaces. AGI32, DIALux and RELux are facilitated with calculation opportunity of required number of luminaires for a defined space. Lumen Method is recommended by Chartered Institution of Building Services Engineers (CIBSE 2002) to calculate required number of luminaires. Also Zonal Cavity method is described as another recommended procedure of luminaire number calculation by Illuminating Engineering Society of North America (IESNA 2000). A comparison of these three computer programs and manual Lumen Method calculation in generating number of luminaires is accomplished to evaluate authenticity of the lighting programs' calculation. This test is conducted in a virtual space for one luminaire type.

*Virtual Space Definition*

The virtual space dimension is 15m(length) x 25m(width) x 2.7m (height). Surface reflectances are: Wall 0.50, Floor 0.20, Ceiling 0.70. Target maintained illuminane level is 500 lux (horizontal) over working plane height (0.75m).

*Computer and Manual Method Calculation*

Luminaire model used for this study is QuattroC 3x18w T26 HF EFL Dif manufactured by Thorn Lighting. Utilisation factors were taken from luminaire manufacturer's catalogue available in Thorn Lighting website. IES photometry of this luminaire is used in all programs to derive required luminaire number in all lighting programs. Results are described in Table 3 and discussed in section 4. Manual calculation by Lumen Method (CIBSE 2002) is described in the following section:

Room definition:
Length (L) = 25 m; Width (W) = 15 m;
Height (H) = 2.7 m
Height of working plane = 0.75m;
$h_m$ (ceiling height – working plane height)= 1.95
Room index (K) = L*W/{(L+W)* $h_m$} = 4.807
Target Illumiance level (E)= 500lux
Area (A) = 375 sqm
Flux output of one lamp (F) = 1350
Number of lamps (n) = 3
Maintenance Factor (MF) = 0.80
Utilisation factors (UF) = 0.58 [derived from datasheet using Room Index]
Numbe of luminaires, n = E x A / F x n x MF x UF
n = 99.78 ~ 100

Table 3: Comparison of luminaire number calculation between different packages and manual methods.

| Name of the program | Calculation method | Luminaire number calculated | Luminaire used in design | Configura-tion | Consider ceiling grid in arrangement |
|---|---|---|---|---|---|
| RELux | Average Indirect Fraction | 102 | 110 | 11x10 | No |
| DIALux | Efficiency Method | Not showing | 110 | 11x10 | No |
| AGi32 | Zonal Cavity Method | 104 | 104 | 13x8 | Yes |
| Manual Calculation | Lumen Method (CIBSE) | 100 | 104 | 13x8 | N/A |

## RESULTS AND DISCUSSIONS

### Illumination Level Calculation

Simulated illuminance values are found very identical with the analytically calculated values, which authenticates the acceptability of lighting calculation for similar types of scenarios. This study only covered two CIE test cases particularly focused on point and area light sources without inter-reflections. Opportunity remains to validate with other test cases and with more complex circumstances. Important aspect noticed is influence of simulation and design parameters on output, like Maintenance Factor, Lamp Loss Factor, etc. which must be considered in real case simulation.

### Luminaire Number Calculation

All programs demonstrated near results compared to Lumen Method calculation, however with few differences. Such as, RELux calculated number of luminaires were 102, suggested to use 110, arranging 11 in one axis and 10 in the other. A manual Lumen method calculation derived required number of luminaires 100 and an arrangement of 104 luminaires (13 x 8) showed both average illuminance (above 500 lux) and uniformity of illuminance (Minimum Illuminance / Average Illuminance, UG1 > 0.5) were able to maintain. This suggestion is also verified by the theories of spacing-to-height ratio (CIBSE 2002). This finding suggests automated arrangement of luminaires is not always precisely ensuring the most efficient power density, however it achieves a near result. Another limitation identified for RELux and DIALux is, not considering ceiling tile grid in automated luminaire arrangements. Even though ability to customize position of luminaires is available but this can raise complexity and include extended manual arrangement process. AGI32 considers 600x600mm or customised ceiling tile grid arrangements for luminaires. However, in this test output failed to meet criteria in some points to meet recommended minimum illuminance and uniformity of illuminance.

None of the programs consider multiple types of luminaires in calculating number and arrangement as they are focused on similar type of output available from manual calculation methods. Newer calculation/optimisation methodologies integrated with computer simulation can play role in supporting lighting design decisions through automated calculation of multiple luminaire types.

### General Discussion of the Packages

AGi32 is found able to handle complex models with its built-in modelling and importing ability from other formats. The program is found as a complete package to provide opportunity to model and calculate lighting, but few limitations found in manipulating and visualizing data in scene, which could be a drawback for analysis and design. Customizable analysis grid of Ecotect provides option to use the grid in several ways for analysis and visualisation. Analysis grid allows multiple data to be saved in one and can provide customized output with graphical representation. The program also can export analysis grid data and allow extended analysis through external programs. The program can be found complicated for some users, as this is a building performance simulation program and contain many other operations other than lighting calculations. Ecotect is also found compatible for modelling and exporting environment geometry for RADIANCE calculation. Using of RADIANCE from Ecotect can be complicated for some users as it is not providing any separate interface for handling RADIANCE materials including luminaire profiles. The use of photometry requires additional use of other RADIANCE commands and demands supplementary knowledge over that.

In DIALux some users can find complications in building complex geometry for its limited model building and 3D object import ability. This is a qualified package for quick calculation with good results with photorealistic image. Relux can be found prospective in defining complex scene description with ability to import 3D geometry from other standard formats. The program also provides limited opportunity to customize material properties and uses RADIANCE to produce photo realistic image.

## CONCLUSIONS

The scene modelling and luminaire defining ability found rational enough for all the packages where DIALux showed possible lack in building complex geometric description. Illuminance level calculation is found within acceptable precisions for all programs in case of simple geometric descriptions and direct lighting. Opportunity still remains for complex geometrical scenarios with inter-reflections to validate. Also it can be suggested that output result is largely dependent on influential physical properties of the environment and light source, so it is advisable to gather sufficient and appropriate space definition and calibrate packages before running simulation. Disability in luminaire number calculation considering ceiling grid can imply limitations or complexity in applying the tools in design practice. More intelligent computation ability is expected in this case. Furthermore application of simulation can be extended to automated luminaire number and arrangement design considering multiple types of luminaires, which demands further study in lighting calculation and optimisation methodologies.

Another aspect distinctly lagging in all programs is manipulation and handling ability of output data. Ecotect shows some advanced features for data manipulation and visualization with wider interactivity while other programs found limited in visualising and exporting output data that is desired for analysis and design. Considering all these AGI32 and Relux found to be potential complete packages for lighting simulation though some modifications are desired. Combination of Ecotect and RADIANCE gives flexibility in defining scene geometry, creating advanced material and visualisation of data; but complexity in operation can make it more applicable in research and development work rather than supporting design decisions.

This study assessed calculation output of illumination level and luminaire number by the programs and used virtual building reference. Lighting environment design involves other aspects like glare evaluation, luminance distribution and moreover energy calculation. Scope remains for future studies to evaluate the programs' ability to manipulate these parameters and in real building case.

## REFERENCES

Ashmore, J. and Richens, P. 2001. "Computer Simulation in Daylight Design: A Comparison." *Architectural Science Review*. 44:1, pp. 33-44.

CIBSE. 2002. *Code for Lighting*. London, Butterworth Heinemann.

Christakou, D. E., Amorim, C. N. D. 2005. "Daylighting

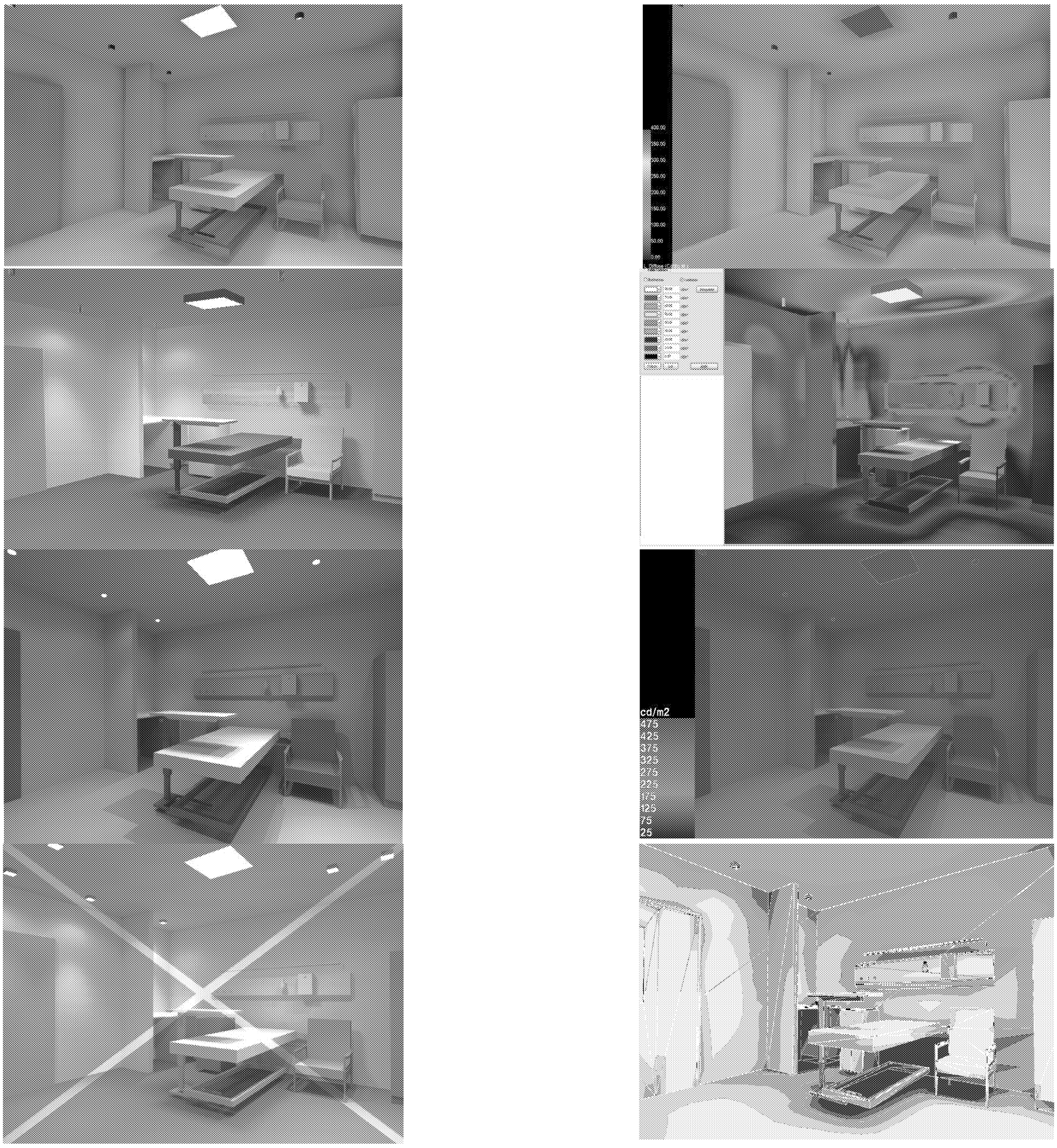


Figure 5: Radiosity and raytrace based rendered images (left) and pseudo colour views of luminance distribution (right) by four packages of a patient room. Image order (from top): AGi32, DIALux, RADIANCE (from Ecotect) and RELux Vision.

Simulation: comparison of Softwares for Architect's Utilization." *Ninth International IBPSA Conference. Montreal. Canada.*

DIAL. Light building software. *DIALux 4 with new improved calculation kernel.* Available at: http://www.dial.de/CMS/English/Articles/DIALux/News/Beitraege_News/Dx4_Rechenkern_eng.pdf (Accessed May 5, 2009).

DOE. 2008. U.S. Department of Energy Building Technologies Program. Building Energy Software Tools Directory. Available at: http://apps1.eere.energy.gov/buildings/tools_directory/subjects.cfm/pagename=subjects/pagename_menu=materials_components/pagename_submenu=lighting_system s (Accessed May 5, 2009)

Galasiu, Anca D., Atif, Morad R. 2002. "Applicability of daylighting computer modelling in real case studies: comparison between measured and simulated daylight availability and lighting consumption." *Building and Environment*. 37, pp. 363-377.

IEA. 2005. "Application of the CIE test cases to assess the accuracy of lighting computer programs." International Energy Agency (IEA) Solar Heating and Cooling Programme Task 31.

INTEGRA. Results of CIE TC.3.33 Tests for Inspirer. Available at: http://www.t-g.de/Download/CIE-lighting-simulation.pdf (Accessed May 5, 2009).

Larson, G. W., Skakespeare, R. 1998. *Rendering with Radiance: The art and Science of Lighting Visualization*. Morgan Kaufman Publishers Inc. San Francisco USA.

Li, D. H. W., and Tsang, E. K. W. 2005. "An analysis of measured and simulated daylight illuminance and lighting savings in a daylit corridor." *Building and Environment*. 40, pp. 973-982.

Ng, E. Y., Poh, L. K., Wei, W., Nagakura, T. 2001. "Advanced lighting simulation in architectural design in the tropics." *Automation in Construction*. 10, pp. 365-379.

Roy, Geoffrey G. 2000. "A Comparative Study of Lighting Simulation Packages Suitable for use in Architectural Design." School of Engineering, Murdoch University.

# MODELING AND SIMULATION OF $CO_2$ ABSORBER COLUMN IN MODELICA

Masoud Najafi[*] Zakia Benjelloun-Dabaghi[†]

## ABSTRACT

Removal of the sour gases Carbon dioxide and Hydrogen sulfide from gas streams has been an important separation process performed via gas absorption process. In the absorption column, the sour gas is absorbed by contact with a liquid solution in which the sour gas is soluble. The absorption column includes several trays allowing mixing the solvent and the gases. In this paper, a model of an absorber column to separate $CO_2$ from $CH_4$ stream will be presented. The absorption column as well as mixing trays are modeled in Modelica. The model is then used to find the concentration of materials over the trays at the equilibrium point.

**KEYWORDS**
Modeling absorption column, Modelica, Equilibrium solution point, Numerical algebraic solver, Scicos

## INTRODUCTION

Carbon dioxide ($CO_2$), Hydrogen sulfide ($H_2S$) and other contaminants are often found in gas streams such as fossil gases. Gases containing $CO_2$ or/and $H_2S$ are commonly referred to as sour gases or acid gases in the hydrocarbon processing industries. These gases are toxic and in many cases undesirable and should be scrubbed off. Gas streams requiring treatment include natural gas, ammonia synthesis gas, gases in petrochemical plants and oil refineries. Almost all sour gas removal processes currently employed involve the absorption of one or more sour components into chemically reactive solvents, mainly aqueous alkanolamines, with the sour gas rich solvents subsequently regenerated thermally, usually by stripping with steam (4, 7, 8). Amine gas treating, which is also known as gas sweetening and sour gas removal, refers to a group of processes that use aqueous solutions of various alkanolamines to remove $CO_2$ or $H_2S$ from gases. An example of reaction formula for the absorption process follows

$$CO_2(gas) + DEA + H_2O \longrightarrow CO_2(absorbed) + ...$$

It is a common process unit used in refineries, petrochemical plants, natural gas processing plants, power plants and other industries. There are many different amines used in gas treating. The most commonly used amines in industrial plants are the alkanolamines:

- Monoethanolamine (MEA)
- Diethanolamine (DEA)
- Methyldiethanolamine (MDEA)

A typical amine gas treating process (see figure 1) includes an absorber unit and a regenerator unit as well as accessory equipments. In an absorber, the downflowing amine solution absorbs $H_2S/CO_2$ from the upflowing sour gas to produce a sweetened gas stream (i.e., a $H_2S/CO_2$-free gas) as a product and an amine solution rich in the absorbed acid gases. The resultant "rich" amine solution is then routed into the regenerator (a stripper with a reboiler) to produce regenerated or "lean" amine that is recycled for reuse in the absorber. The stripped overhead gas from the regenerator is concentrated $H_2S/CO_2$.

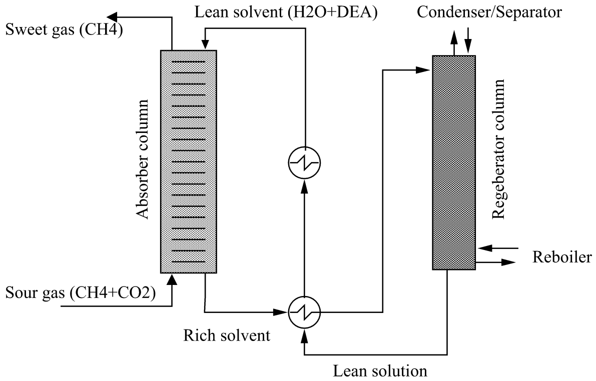


Figure 1: Diagram of a typical amine treating process used in industrial plants.

In this paper the Modelica model of the absorber column is presented. This model is in particular used to obtain the concentration of species over the trays at the equilibrium point. In this paper, strippers, reboiler and condensation units are not modeled. In the first part of the paper the absorber column process model will be described. The model is developed from the principles of diffusion from vapor to liquid taking into account the chemical reactions between the $DEA$ and $CO_2$. The Modelica model and the simulation are included in section 3.

[*] INRIA-Rocquencourt, Domaine de Voluceau, 78153, Le Chesnay, France. masoud.najafi@inria.fr

[†] IFP, 1 avenue de Bois-Préau, 92852, Rueil-Malmaison, France. zakia.benjelloun-dabaghi@ifp.fr

## ABSORBER COLUMN

The details of the $CO_2$ absorption into an amine solution in an absorption column are quite complex. There are many references about the chemistry involved in the process and models including mass transfer mechanisms and chemical reactions kinetics in the literature, interested readers are referred to, *e.g.*, (3, 2, 9, 5).

In absorber columns, the gas and liquid phases flow in a counter-current pattern across a finite number of physical trays (or a structured packing) where the two liquid and gas phases are mixed thoroughly, see Fig. 2. Furthermore, the total molar flows of the liquid and gas phases are nearly constant in an adiabatic column section with no external feed.

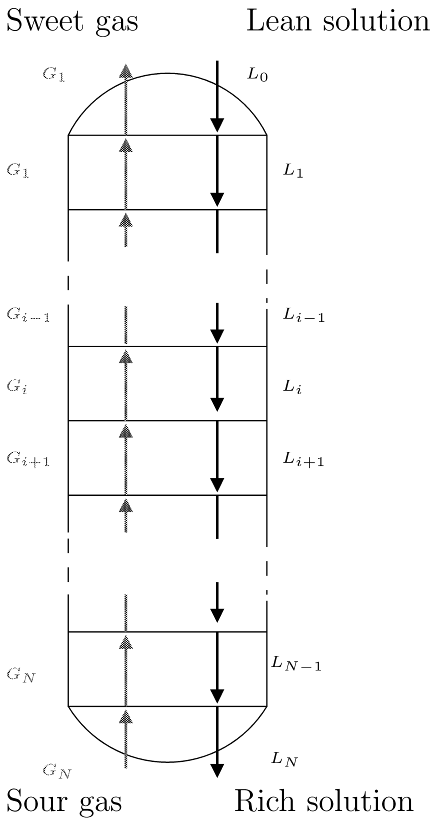


Figure 2: Counter-current absorber column.

The absorber column has two inlet and two outlet streams.

- Bottom inlet: The sour gas containing $CH_4$ and $CO_2$ are pumped in.
- Head inlet: The lean solution containing DAE and $H_2O$ are pumped in.
- Bottom outlet: The rich solution containing solution and absorbed $CO_2$ flows out.
- Head outlet: sweet gas (pure $CH_4$) streams out.

In the model used at IFP, the inlet flow of solution and sour gas as well as molar fraction of species at inlets are specified. The conservation of mass and heat transfer laws have been applied. Henry's law has been used to model the vapor phase equilibrium of the $CO_2$. Chemical reactions between $CO_2$ and $DEA$ were included in the model for chemical absorption. Liquid and vapor energy balances were used to compute the liquid and vapor temperature, respectively.

In this model, we consider the absorption of the $CO_2$ mixed with $CH_4$ in an aqueous solution of DEA. $CO_2$ is absorbed in the aqueous solution down-falling from the upper tray and reacts with the DAE.

From the global mass balance over the $i$ $^{th}$ tray we can write:

$$\begin{array}{rcll} L_i & = & L_{i-1} + \phi_i & Liquid \;\; phase \\ G_i & = & G_{i+1} - \phi_i & Gas \;\; phase \end{array}$$

where $\phi_i$ is the transfer rate of a component between gas and liquid phases over the $i$ $^{th}$ tray.

From and the partial mass balance over the $i$ $^{th}$ tray for liquid and gas phases the following equations can be obtained:

$$\begin{array}{rcll} L_{i-1}\; x^j_{i-1} & = & L_i\; x^j_i - \phi^j_i & Liquid \;\; phase \\ G_{i+1}\; y^j_{i+1} & = & G_i\; y^j_i + \phi^j_i & Gas \;\; phase \end{array}$$

where $x^j_i$ and $y^j_i$ are the molar fraction of species $j$ in liquid and gas phases over the $i$ $^{th}$ tray, respectively.

The transfer rate of the $j$ $^{th}$ species between gas and liquid phase over the $i$ $^{th}$ tray ($\phi^j_i$) is described by

$$(\frac{1}{k_g} + \frac{H^j_i}{k_l\; E^j_i})\phi^j_i = (y^j_{i+1}P - H^j_i\;\; C^j_i).a$$

where $a$, $k_l$ and $k_g$ are tray's physical parameters. $E^j_i$ is a function of concentration of DEA and the temperature. $H^j_i$ is the constant of Henry of the $j$ $^{th}$ species over the $i$ $^{th}$ tray. $C^j_i$ is the concentration of the $j$ $^{th}$ species over the $i$ $^{th}$ tray. $P$ is the pressure which is assumed to be constant. The concentration of $DAE$, $CO_2$, and Henry constants are computed as a function of $w^{DAE}_i$ and $\alpha^{CO_2}_i$ where $w^{DAE}_i$, the mass fraction of amine in the lean solution, is given by:

$$w^{DAE}_i = \frac{k_1 x^{DAE}_i}{k_1 x^{DAE}_i + k_2 x^{H_2O}_i}$$

where $k_1$ and $k_2$ are constant parameters.

$\alpha^{CO_2}_i$, the fraction rate of the absorbed $CO_2$ in the rich solution, is described by:

$$\alpha^{CO_2}_i = \frac{x^{CO_2}_i}{x^{DAE}_i}$$

Due to the absorption of $CO_2$ over trays, the temperature of trays changes. Assuming that the temperature of gas and outgoing liquid over a tray are equal, we get the following equation describing the temperature over the $i^{th}$ tray:

$$L_i \; T_i = L_{i-1} \; T_{i-1} + \frac{\Delta H_{CO_2-DAE}}{C_p} \; \phi_i^{CO_2}$$

where $\Delta H_{CO_2-DAE}$ is the enthalpy of the reaction and is considered constant.

## MODELICA MODEL OF THE ABSORPTION COLUMN

In order to model the process in Modelica, three blocks have been defined: a block for the head of the absorption column (*head*), a block for the *tray* and a block for the bottom of the column (*bottom*). The *head* block is used to specify the lean solvent flow rate, molar fraction of $CO_2$, $DEA$ and $H_2O$ in the input lean solution (liquid). The *tray* block models a tray in the column. Several parameters of the tray and constants used in the model can be defined. The column bottom block specifies the sour gas flow rate and molar fraction of $CO_2$, $CH_4$ in the input sour gas.
In order to connect the Modelica blocks, a Modelica connector carrying the following variables has been used.

- Gas: Gas flow rate
- phi: Transferred material
- T: Temperature
- Liq: Liquid
- yCO2, yDEA, yH2O, yCH4: Molar fractions in gas
- xCO2, xDEA, xH2O, xCH4: Molar fractions in liquid

The Modelica model has been implemented in the Scicos[1] software (1). The Modelica model of the absorption column developed in Scicos is displayed in Figure 3. In this model the column is composed of 20 trays.
Scicos uses a special GUI to solve the algebraic equations resulting from the Modelica equations. This GUI allows the user to try different algebraic solvers for solving the equation (6). The GUI used for solving the steady state solution of the model depicted in Fig. 3 is given in Fig. 4. The result can be saved in an XML file to be used later as initial state values for future simulations or used by other tools.
In this model, we are interested in finding equilibrium point molar fractions of $CO_2$ and $CH_4$ over each tray to see the efficiency of the absorption column. The concentration of $CO_2$ in the sour gas stream at the column bottom is 15 $mol/m^3$. At this point, the molar fraction of $CO_2$ and $CH_4$ are 0.1 and 0.9, respectively.
The developed model is not dynamic and since the system is purely algebraic, the equilibrium point is found by solving a nonlinear system of equations. In order to solve the system at steady state, Scicos uses a special GUI to solve the algebraic equations resulting from the Modelica model. This GUI provides the user several numerical algebraic solvers for solving the system of equations (6). Two methods for initializing a Modelica model is provided, one of which is solving the system at steady state. An screen shot of this GUI used for solving the model of Fig. 3 is given in Fig. 4. The result can be saved in an XML file to be used later as initial state values for future simulations or used by other tools. In Fig. 5, the molar fraction of $CO2$ and $CH4$ in the sour gas stream is depicted as a function of tray number, it can be seen that at the column head the molar fraction of $CO_2$ in the gas is negligible, *i.e.,* $CO_2$ has been absorbed in the solvent and $CH_4$ is pure.

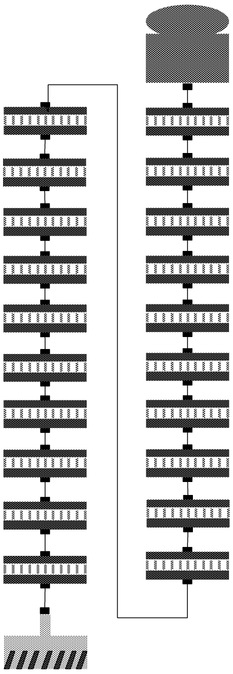
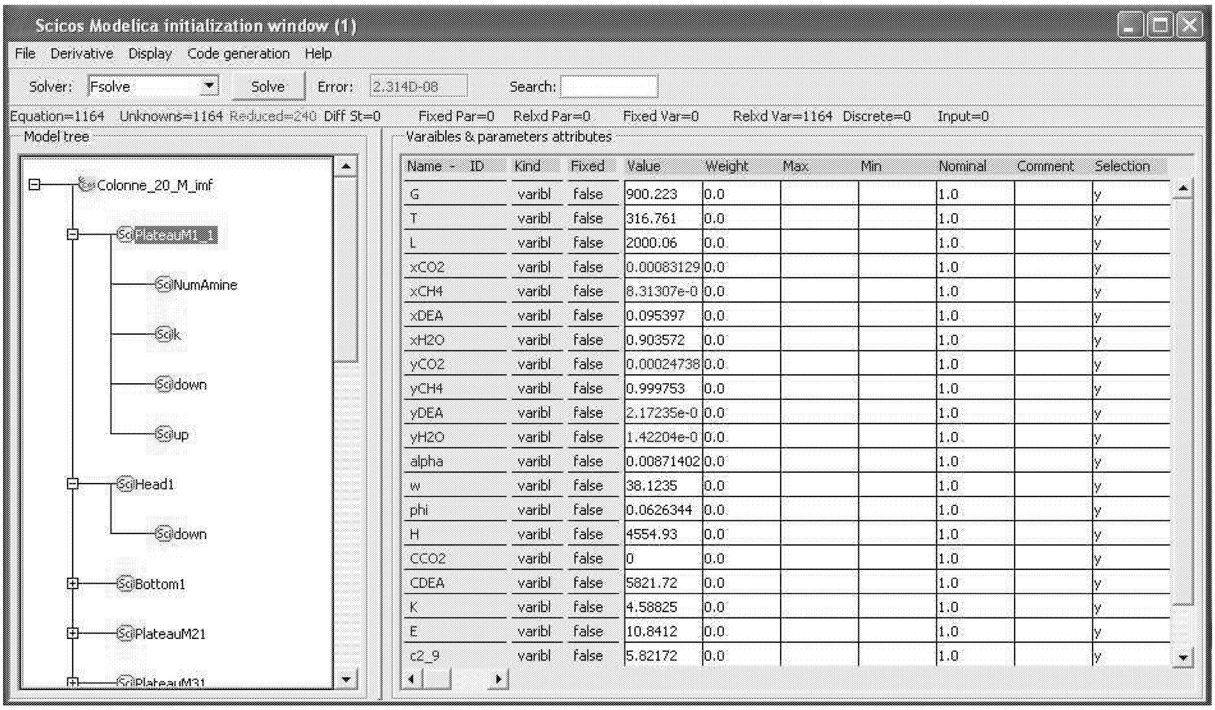


Figure 3: Modelica model of the absorption column composed of 20 trays modeled in Scicos

The Modelica model can be used to observe the behavior of the absorption column. Several parameters such as composition of the head and bottoms products at various feed rates and inlet compositions can be computed. An important feature of the Modelica model is the fact that this model can be easily used for parameter estimation and parameter sizing. For example estimating required input flow rate to get the desired output sweet gas flow rate.

In this work, we assumed a uniform and constant concentration of species over trays. In a more detailed model, this should be changed and instead a model with

[1] Scicos is a free and open-source simulation software developed at INRIA and is available at *www.ScicosLab.org*.



Solver: Fsolve

Error: 2.314D-08

Search:

Equation=1164 Unknowns=1164 Reduced=240 Diff St=0 Fixed Par=0 Relxd Par=0 Fixed Var=0 Relxd Var=1164 Discrete=0 Input=0

Varaibles & parameters attributes

| Name - ID | Kind | Fixed | Value | Weight | Max | Min | Nominal | Comment | Selection |
|---|---|---|---|---|---|---|---|---|---|
| G | varibl | false | 900.223 | 0.0 | | | 1.0 | | y |
| T | varibl | false | 316.761 | 0.0 | | | 1.0 | | y |
| L | varibl | false | 2000.06 | 0.0 | | | 1.0 | | y |
| xCO2 | varibl | false | 0.00083129 | 0.0 | | | 1.0 | | y |
| xCH4 | varibl | false | 8.31307e-0 | 0.0 | | | 1.0 | | y |
| xDEA | varibl | false | 0.095397 | 0.0 | | | 1.0 | | y |
| xH2O | varibl | false | 0.903572 | 0.0 | | | 1.0 | | y |
| yCO2 | varibl | false | 0.00024738 | 0.0 | | | 1.0 | | y |
| yCH4 | varibl | false | 0.999753 | 0.0 | | | 1.0 | | y |
| yDEA | varibl | false | 2.17235e-0 | 0.0 | | | 1.0 | | y |
| yH2O | varibl | false | 1.42204e-0 | 0.0 | | | 1.0 | | y |
| alpha | varibl | false | 0.00871402 | 0.0 | | | 1.0 | | y |
| w | varibl | false | 38.1235 | 0.0 | | | 1.0 | | y |
| phi | varibl | false | 0.0626344 | 0.0 | | | 1.0 | | y |
| H | varibl | false | 4554.93 | 0.0 | | | 1.0 | | y |
| CCO2 | varibl | false | 0 | 0.0 | | | 1.0 | | y |
| CDEA | varibl | false | 5821.72 | 0.0 | | | 1.0 | | y |
| K | varibl | false | 4.58825 | 0.0 | | | 1.0 | | y |
| E | varibl | false | 10.8412 | 0.0 | | | 1.0 | | y |
| c2_9 | varibl | false | 5.82172 | 0.0 | | | 1.0 | | y |

Figure 4: Screenshot of the initialization GUI used for solving the model of Fig. 3.

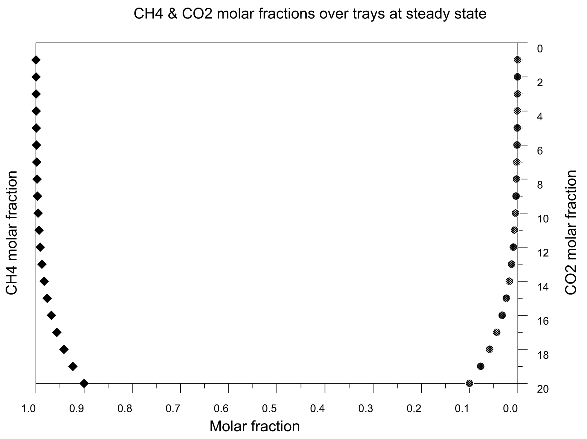


Figure 5: Molar fraction of $CO_2$ and $CH_4$ along the absorption column (tray #1 is the head of the column)

distributed concentrations be used. Based on the model explained in this paper, a more elaborated model is developed where the concentration of species over the tray is described by partial differential equations, see(10).

## CONCLUSION

The process of the absorption of $CO_2$ into Diethanolamine by in absorption columns has been modeled in this paper with the Modelica language. The model is in particular used to obtain the equilibrium point of the solution. This model is used as a base for developing more elaborated models with dynamic states for the absorption column.

## REFERENCES

[1] S. L. Campbell, J.P. Chancelier, R. Nikoukhah, Modeling and simulation in Scilab/Scicos, Springer Verlag publishing, 2005.

[2] Danckwerts, P.V., Sharma, M.M. 'The absorption of Carbon Dioxide into solutions of alkalis and amines', The Chemical Engineer, pp 244-280, 1966

[3] Greer et al., A dynamic model for the de-absorption of carbon dioxide from Monoethanolamine solution, SIMS 2008, Oslo, Norway.

[4] A. Kohl, R. Nielsen, 'Gas Purification', Fifth Edition, 1997

[5] H.M. Kvamsdala, J.P. Jakobsena, K.A. Hoffb, 'Dynamic modeling and simulation of a CO2 absorber column for post-combustion CO2 capture', Chem Eng and Proc: Process Intensification Volume 48, Issue 1, Jan 2009, pp 135-144.

[6] M. Najafi, R. Nikoukhah, "Initialization of Modelica Models in Scicos", Modelica conference, Germany, 2008.

[7] J. Polasek, and J. Bullin, 'Selecting Amines for Sweetening Units', Gas Processors Asso. Regional Meeting, Sept. 1994.

[8] D. Wallace,'Capture and Storage of CO2 what needs to be done?', November 2000, Available from http://www.earthscape.org/r1/iea05/

[9] Versteeg G. F., Van Dijck L. A. J., Van Swaaij W. P. M., 'On the kinetics between CO2 and alkanolamines both in aqueous and non-aqueous solutions, an overview', Chem. Eng. Comm. 144, 113-158, 1966

[10] Najafi M., Benjelloun-Dabaghi Z., 'Simulation of PDE's in Modelica, Application to Absorber Column',ESM2009 Conf., UK, 2009.

# SIMULATION OF PDES IN MODELICA, APPLICATION TO ABSORBER COLUMN

Masoud Najafi[*] Zakia Benjelloun-Dabaghi[†]

## ABSTRACT

PDEs are quite common in modeling physical systems with distributed and lumped parameters. In order to simulate numerically such models, PDEs are first converted to ODEs, then simulated using ODE solvers. There are simple and straightforward methods to convert PDE to ODE such as the finite difference method. Although simple, these methods result in big, stiff and sparse matrices. In this paper we use the orthogonal collocation method to convert PDE to ODE. The interesting property of the orthogonal collocation method is to generate smaller ODE, *i.e.*, with a few dicretization points compared to other methods. As an application, an absorption column has been modeled. The detailed model of the absorption column includes PDEs. In this paper, the orthogonal collocation method has been used to model the process in Modelica. The Modelica model is then simulated with the Scicos simulation software.

**KEYWORDS**
Modelica, Partial Differential Equation, Absorption Column, Scicos

## INTRODUCTION

PDEs arise in models of numerous scientific and industrial applications. Although simulate, analysis and prediction of the behavior of dynamical systems with distributed and lumped parameters are very important in industry, most existing modeling languages and simulators are only suited to model and simulate lumped parameter systems. In fact, many interesting engineering systems such as most mechanical systems, chemical process with mass and heat transfer are intrinsically distributed in nature. It means that their properties exhibit spatial as well as temporal variations. The resulting set of equations for these types of models may be viewed as a combination of lumped and distributed parameter systems which is called Partial Differential Equations (PDE) subjected to initial and boundary conditions.

The simulation of a PDE is not a trivial task and needs a discretization of the space at many points to have a reasonable accuracy. There are large number of numerical methods for the simulation of PDE systems. Numerical methods may be based on :

- Method of lines,
- Finite difference methods,
- Weighted residuals methods,
- Finite Element/Finite Volume methods,
- Adaptive/Moving grid methods.

These methods convert a PDE into sets of ODEs involving space discretization. The main advantage of this method is converting the PDE into an ODE or a DAE to use available appropriate numerical solvers, *e.g.*, `Dassl` (15) or `Sundilas` (19, 20). But In general ODE/DAE solvers cannot easily control and estimate the effects of the space discretization error on the general numerical scheme. The main disadvantage of these methods are the large size of the resulting ODE/DAE system obtained after discretization and the rigidity of the fixed grid discretization scheme. In many applications such as process control or fault detection, a compromise between the low model complexity and the high solution accuracy need to be found. For this reason, after the global resolution method with higher accuracy, a functional approximation method is used to obtain a state representation with a low finite dimension with a satisfying accuracy for control purposes.

There are, however, some spectral methods to discretize a PDE using only a few number of discretization points. This method of space discretization is called the orthogonal collocation method. The collocation method is a functional approximation method to convert PDE to ODE with low finite number of discretization points. Some precautions concerning the choice of base functions and others parameters respected, we can use the collocation method as an efficient and powerful method. Moreover, this method is conservative for mass and heat balances, since at discretization points, the PDE is satisfied (9, 7). In this paper, we apply this method to simulate a PDE resulting from modeling an absorption column. First, we give a brief review on the orthogonal collocation method. Then the absorption column as a chemical process will be presented. This process which includes PDEs, will be modeled in Modelica. The Modelica model is then simulated in Scicos.

[*] INRIA-Rocquencourt, Domaine de Voluceau, BP 105, 78153, Le Chesnay, France. `masoud.najafi@inria.fr`

[†] IFP, 1 & 4, avenue de Bois-Préau, 92852, Rueil-Malmaison, France. `zakia.benjelloun-dabaghi@ifp.fr`

## MATHEMATICAL BACKGROUND

Let consider the following PDE.

$$\frac{\partial\phi(x,t)}{\partial t}=a\frac{\partial^2\phi(x,t)}{\partial x^2}+b\frac{\partial\phi(x,t)}{\partial x}+f(x,t) \quad (1)$$

where $\phi(x,t)$ denote a physical quantity in a 1-D environment over time and $a$ and $b$ are constants. In order to solve 1, two other information are needed; The initial values of states, *i.e.*,

$$\phi(x,t=0)=\phi_0(x), \quad \texttt{for} \;\; 0\le x\le L.$$

And boundary conditions which may be defined in three forms:

- Dirichlet problem, *i.e.*, the value of $\phi$ is defined at the boundary $B$,

$$\phi(x,t)|_{x=B}=f(B,t)$$

- Neumann, *i.e.*, the normal derivative of $\phi$ is a specified at the boundary $B$,

$$\frac{\partial\phi}{\partial x}|_{x=B}=f(B,t)$$

- Cauchy (Robins), *i.e.*, a mixed condition defines $\phi$ at the boundary $B$

$$\alpha\phi(x,t)+\beta\frac{\partial\phi(x,t)}{\partial x}=f(B,t)$$

## THE COLLOCATION METHOD

The collocation method was developed originally as a stable, predictable, and simple to implement pseudo-spectral technique. Because of its reliability, it has become a standard method for solving boundary-value problems by polynomial trial function expansions. This method permits to discretize a PDE with selection of only 4 to 7 points in the region, comparing to 10 to 20 points in the finite difference method.

The formulation of this method is based on choosing a set of trial functions from an orthogonal polynomial sequence (9, 10, 8).

Similar to the finite difference method, in collocation methods $N$ points are selected, (*i.e.*, $x_i$) and the region is divided into $N+1$ segments. Then, $\phi(x,t)$ is approximated by a Lagrange interpolation polynomial $\hat{\phi}(x,t)$ of order $N+1$ using $x_i$ as interpolation points. Then, the approximated partial derivatives at $x_i$ (corresponding derivatives of $\hat{\phi}$) are inserted in the PDE to satisfy the PDE **only** at collocation points ($x_i$). In order to implement this method in a 1-D region with $N$ collocation points, $\phi$ and its partial derivatives are approximated as linear combinations of basis functions $L_i$, *i.e.*,

$$\phi\approx\hat{\phi}=\sum_{i=1}^{N}\hat{\phi}_i(t)\cdot L_i(x) \quad (2)$$

$$\Rightarrow\frac{\partial\phi}{\partial x}\approx\frac{\partial\hat{\phi}}{\partial x}=\sum_{i=1}^{N}\hat{\phi}_i(t)\cdot\frac{dL_i(x)}{dx} \quad (3)$$

$$\Rightarrow\frac{\partial^2\phi}{\partial x^2}\approx\frac{\partial^2\hat{\phi}}{\partial x^2}=\sum_{i=1}^{N}\hat{\phi}_i(t)\cdot\frac{d^2L_i(x)}{dx^2} \quad (4)$$

$$\Rightarrow\frac{\partial\phi}{\partial t}\approx\frac{\partial\hat{\phi}}{\partial t}=\sum_{i=1}^{N}\frac{d\hat{\phi}_i(t)}{dt}\cdot L_i(x) \quad (5)$$

$L_i(x)$ are the $N+1$ linearly independent basis functions of the $N^{th}$ order Lagrange polynomial. They are determined by the $N+1$ collocation points $x_i$ as follows:

$$L_i(x)=\prod_{j=1,\;i\neq j}^{N}\frac{x-x_j}{x_i-x_j}=a_Nx^N+\cdots+a_1x+a_0 \quad (6)$$

The coefficients $a_k$ can be computed from the known points $x_i$. Since the PDE is to be satisfied only at the collocation points and the Lagrange polynomials (6) have the following property

$$\begin{cases} L_i(x_j) &= 1 \quad \texttt{for} \;\; i=j\\ L_i(x_j) &= 0 \quad \texttt{for} \;\; i\neq j,\end{cases}$$

then, we can simplify the equations (2-5). There are some interesting remarks to be noted in this method:

- The coefficients $\hat{\phi}_i(t)$ of the linear combination (2) correspond to the values of $\hat{\phi}(x,t)$ at the collocation points $x_i$, because we use Lagrange polynomials. This means that the derivatives $\frac{\partial\hat{\phi}}{\partial x}$, $\frac{\partial^2\hat{\phi}}{\partial x^2}$ and the state $\hat{\phi}$ satisfies the PDE only at a given collocation point. At other points the PDE will not be satisfied. Hence, the states $\hat{\phi}_i$ cannot be exact neither the derivatives.

- The discretization does not depend on the PDE, but only on the choice of collocation points.

- The coefficients of $\hat{\phi}_i(t)$ in (3-5) are constant and can be computed in advance using the placement of $x_i$ points.

- $\hat{\phi}_i(t)$ are the only time varying variables. Thus, the number of ODE variables of the ODE system is $M=(N+1)N_s$. Where $N_s$ is the dimension of the state vector $\phi$.

In order to simplify the equations,we use a normalized dimensionless variable $z$ to replace the $x$-coordinate in the PDE of (1), defined by $z=x/L$, so that as $x$

varies over $[0, L]$, and $z$ correspondingly varies over $[0, 1]$. Therefore (1) becomes:

$$\frac{\partial \phi(Lz,t)}{\partial t} = \frac{k_1}{L^2}\frac{\partial^2 \phi(Lz,t)}{\partial z^2} + \frac{k_2}{L}\frac{\partial \phi(Lz,t)}{\partial z} + f(Lz,t) \quad (7)$$

## ORTHOGONAL COLLOCATION METHOD

The most difficult decision in the collocation method is the placement of the collocation points. In (12), it is showed that the integration of differential equations gives the best result if the collocation points are placed at the zeros of an orthogonal polynomial. Cho and Joseph (11) placed the collocation points at the zeros of Jacobi polynomials defined by (8).

The interpolation with the Lagrange polynomials tends to oscillating curves yielding a very bad approximation, if the collocation points are not well chosen. In order to avoid this problem, the basis functions of the Lagrange polynomial should be a set of orthogonal functions in the interval $0 \leq z \leq 1$ with respect to some weighting function $w(z)$. When the interpolation points $z_i$ are chosen as roots of Jacobi polynomials, the Lagrange polynomials are orthogonal in the sense of

$$\begin{aligned} &\int_0^1 w(z) L_i(z) L_j(z) dz = 0 \quad \texttt{for} \quad i \neq j \\ &w(z) = (1-z)^\alpha z^\beta \end{aligned} \quad (8)$$

The Jacobi polynomial of order $N$ is defined as:

$$P_N^{(\alpha,\beta)} = \sum_{i=0}^{N} (-1)^{N-i} \gamma(N,i) z^i$$

The function $\gamma(N,i)$ can be computed recursively as follows:

$$\begin{aligned} &\gamma(N,i) = \frac{(N-i+1)(N+i+\alpha+\beta)}{i(i+\beta)} \gamma(N,i-1) \\ &\gamma(N,0) = 1 \end{aligned} \quad (9)$$

The parameters $\alpha$ and $\beta$ can be used to influence the positioning of collocation points ($z_i$). This results from the weighting function $w(z) = (1-z)^\alpha z^\beta$, *i.e.*,

| | | | |
|---|---|---|---|
| $\alpha$ *small* | : | `higher density toward` | $z = 1$ |
| $\beta$ *small* | : | `higher density toward` | $z = 0$ |
| $\alpha = \beta$ | : | `symmetric distribution` | |

Their default choice of the parameters for the Jacobi polynomial ($\alpha = 0.5,\ \beta = 0.5$) results in a symmetrically spaced points respecting two endpoints. The collocation points can move toward either end of the collocation region by adjusting $\alpha$ and $\beta$ parameters. This can be used to get more accurate at certain points of the collocation region. More about this issue can be found in (13).

The roots of the Jacobi polynomials never include 0, and using $\alpha$= -1 (yields a root at 1) might be a bad choice for given PDE. Hence, the boundary values have to be extrapolated using (2) for Dirichlet conditions or (3) for Neumann conditions. Alternatively, one can use additional collocation points at those boundaries that have boundary conditions in order to avoid extrapolation. Note that by doing this, some of the basis functions are not orthogonal to the others (9, 10, 8).

Both parameters $\alpha$ and $\beta$ are considered as optimization parameters for the collocation point selection. The best choice depends on the model. It has been shown that for nonlinear systems, it is better to place the collocation points at regions with higher nonlinearity (11). In fact, it is showed that the model accuracy depends more on the collocation point location than on their number; inappropriate values of $\alpha$, $\beta$ and $N$ lead to numerical instabilities (7).

## APPLICATION: ABSORPTION COLUMN

Carbon dioxide ($CO_2$), Hydrogen sulfide ($H_2S$), and other contaminants are often found in gas streams such as fossil gases. Gases containing $CO_2$ or/and $H_2S$ are commonly referred to as sour gases or acid gases in the hydrocarbon processing industries. These gases are toxic and in many cases undesirable and should be scrubbed off. Removal of the sour gases $CO_2$ and $H_2S$ from gas streams has been an important separation process. Almost all sour gas removal processes currently employed involve the absorption of one or more sour components into chemically reactive solvents, mainly aqueous alkanolamines, with the sour gas rich solvents subsequently regenerated thermally, usually by stripping with stream (3, 4, 16). Amine gas treating which is also known as gas sweetening and sour gas removal refers to a group of processes that use aqueous solutions of various alkanolamines to remove $CO_2$ or $H_2S$ from gases. There are many different amines used in gas treating. The most commonly used amines in industrial plants are the alkanolamines, Monoethanolamine (MEA), Diethanolamine (DEA) and Methyldiethanolamine (MDEA).

A typical amine gas treating process (see Figure 1) includes an absorber unit and a regenerator unit as well as accessory equipments. In an absorber, the down-flowing amine solution absorbs $H_2S/CO_2$ from the up-flowing sour gas to produce a sweetened gas stream, (*i.e.*, a $H_2S/CO_2$-free gas) as a product and an amine solution rich in the absorbed acid gases. The resultant "rich" amine solution is then routed into the regenerator (a stripper with a reboiler) to produce regenerated or "lean" amine that is recycled for reuse in the absorber. The stripped overhead gas from the regenerator is concentrated $H_2S/CO_2$.

The details of the $CO_2$ absorption into an amine solution in an absorption column are quite complex. There are many references about the chemistry involved in the process and models including mass transfer mechanisms

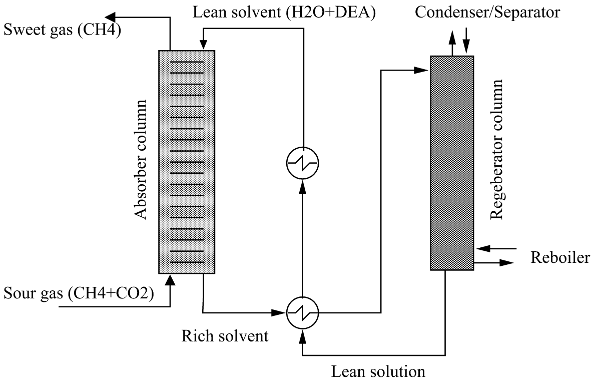


Figure 1: Diagram of a typical amine treating process used in industrial plants

and chemical reactions kinetics in the literature, interested readers are referred to, *e.g.*, (6, 18, 17, 14).

In absorber columns, the gas and liquid phases flow in a counter-current pattern across a finite number of physical trays (or a structured packing) where the two liquid and gas phases are mixed thoroughly.

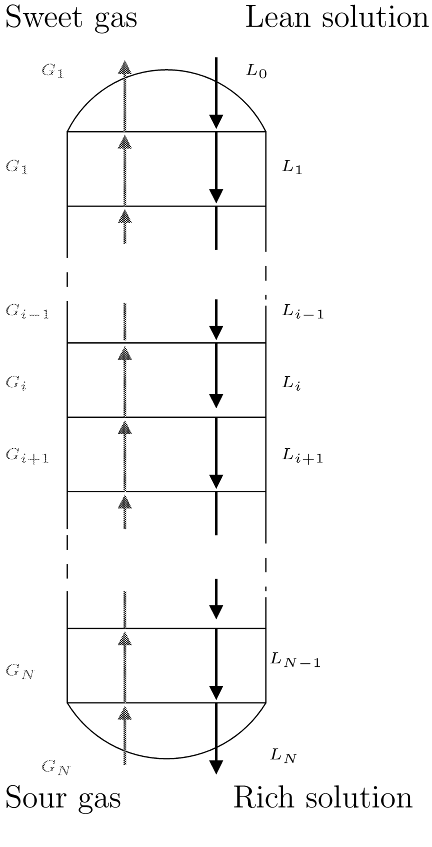


Figure 2: Counter-current absorber column

In this paper, the model of the absorber column with lumped parameter trays developed in (21) is used to develop a more elaborated model with concentration of species over tray described by PDE's.

We consider the case of absorption of $CO_2$ in an aqueous solution of MDAE. The chemical reactions over trays are mainly as follows:

$$\begin{array}{lcl} CO_2 + MDEA + H_2O & \leftrightarrows & MDEAH^+ + HCO_3^- \\ CO_2 + OH^- & \leftrightarrows & HCO_3^- \\ CO_3^{2-} + H_2O & = & HCO_3^- + OH^- \\ MDEA + H_2O & = & MDEAH^+ + OH^- \end{array}$$

Based on the theory of thin film and taking into account the gradient of electrostatic potential for the diffusion of ionic species, the system of equations describing the behavior of concentrations is given by the Nernst-Planck equations as follows :

$$\begin{aligned} \frac{\partial C_i(x,t)}{\partial t} &= D_i \frac{\partial^2 C_i(x,t)}{\partial x^2} - z_i D_i \frac{F}{RT} \frac{\partial \phi(x,t) C_i(x,t)}{\partial x} \\ &\quad + G_i(x,t) \end{aligned} \tag{10}$$

where $i$ represents all species *i.e.*, $CO_2$, $MDEA$, $MDEAH^+$, $HCO_3^-$, $CO_3^{2-}$ and $OH^-$. $D_i$ represents the diffucion coefficient of the species $i$.

$\phi(x,t)$ is the gradient of the electrostatic potential which couples the diffusion of ionic espies. This term plays an important role where the coefficients of diffusion of ionic species are different. Assuming that dynamic electroneutrality and using the Nernst-Planck equations, $\phi(x,t)$ can be defined as a function of concentrations of ions and their diffusion coefficients.

$$\phi(x,t) = \frac{RT \displaystyle\sum_{i=1}^{M} z_i \ D_i \ \frac{\partial C_i}{\partial x}}{F \displaystyle\sum_{i=1}^{M} z_i^2 \ D_i \ C_i}$$

$M$ is the number of all species (here $M = 6$).

$G_i(x,t)$ is the production rate of species $i$ , *i.e.*, reaction terms:

$$G_i(x,t) = \sum_{j=1}^{N} \lambda_{i,j} \ k_j \ \prod_{k=1}^{M} C_k^{\beta_{k,j}}$$

where $j$ represents non-ionic components and $N$ is their number (here $N = 5$). For a ionic species (in our case $OH^-$), one equation of 10 is very often replaced by an algebraic equation which is called the electroneutrality equation:

$$\sum_{i=1}^{M} z_i \ C_i = 0$$

Thus the behavior of the absorption column is in fact a Differential Algebraic Equation (DAE).

## INITIAL CONDITIONS

For gas species (here $CO_2$), we use a linear profile considering that there is an equilibrium between gas and liquid at the surface. Furthermore, the concentration of gas at the boundaries of the film is identical to the concentration of the liquid mass.
The concentration of gas ($CO_2$) is

$$C_i(x,0) = (1 - \frac{x}{L})\frac{P_i}{H_i} + C_{i,mz}\frac{x}{L}$$

and the concentration of non-gas species (all except $CO_2$) is

$$C_i(x,0) = C_{i,mz}$$

where $C_{i,mz}$ is the concentration at the mixing zone over the tray.

## BOUNDARY CONDITIONS

• At $x = 0$
At the gas-liquid interface, we have distinguished the Transferred species (gas species) from others. For gas species ($CO_2$), we have adopted a boundary condition by considering the flow continuity of Transferred species at the gas-liquid interface:

$$K_{G,i}(P_i - H_i C_i) = -D_i(\frac{\partial C_i}{\partial x})|_{x=0}$$

which can be rewritten as follows.

$$C_i(0,t) = \frac{D_i}{K_{G,i}H_i}\frac{\partial C_i}{\partial x}|_{x=0} + \frac{P_i}{H_i}$$

It is important to note that we have assumed that the temperature is constant all over each tray.
For non-gas species we have

$$\frac{\partial C_i}{\partial x}|_{x=0} = 0$$

This equation indicates that there is no flow for non Transferred species.

• At $x = L$
In the mixing zones of liquid, we assume that there is a chemical equilibrium between all species of the solution. Thus we have:

$$C_i(L,t) = C_{i,mz}$$

## MODEL OF THE ABSORPTION COLUMN IN MODELICA

The dynamic model of the absorption column developed at IFP tries to be more accurate than the model given in (21). It means that they may be used in various configurations such as abrupt changes, start-up and shutdown configurations,... These models should further be used for simulation, including parameter estimation, even in the near future for fast real time computing. It may involve in realistic environment the necessity of state events detection, treatment of internal and external discontinuities. The Modelica language as it is now does not allow a direct conversion of PDEs into ODE, thus the conversion is done by the user and the resulting ODE/DAE system is then modeled in Modelica and then solved by any Modelica simulator.
The Modelica model can be used to observe the behavior of the absorption column. Several parameters such as composition of the head and bottoms products at various feed rates and inlet compositions can be computed. An important feature of the Modelica model is the fact that this model can be easily used for parameter estimation and parameter sizing at steady state. For example estimating required input flow rate to get the desired output sweet gas flow rate.
Modelica provides an easy way to develop chemical process models. Using the Modelica language provides many advantages over classical methods of modeling and simulation. The model may be used in various configurations such as abrupt changes, start-up and steady-state simulation. Furthermore, the model can be used in parameter estimation, even in the near future for fast real time computing. The symbolic nature of the language allows extraction of several information from the model that helps the numerical solver to simulate the system more efficiently, e.g., state events and internal and external discontinuities are automatically handled by the solver. Another important feature of Modelica is the fact that, the model is independent of the simulator tool and theoretically can be simulated with any Modelica simulator.
The Modelica compiler of Scicos, as it is today, does not allow a direct conversion of PDEs into ODE. Thus the conversions has been done by the user and the resulting system is simulated by Scicos.

## SIMULATION RESULT

In order to test and validate the new Modelica model, we try to find the concentration of species over a tray at the equilibrium state. In order to start the simulation, the concentration of species at initial time over the tray is needed. In this simulation, we set the concentration of $CO_2$ at the gas-liquid interface (*i.e.,* $x = 0$) to 10 $mol/m^3$. The concentration of other species (liquid species) are $C_i(x, t = 0) = C_{i,mz}$. The initial concentration of $CO_2$ is chosen so that the effect of gradient of electrostatic potential ($\psi$) be important in the evolution of concentrations. In Fig. 3 the time evolution of the concentration of $CO_2$ over the tray is depicted. In Fig. 4, the concentration of $CO_2$ at the equilibrium state over the tray as a function of space discretization indices is given.
The results we obtained with the new model were the

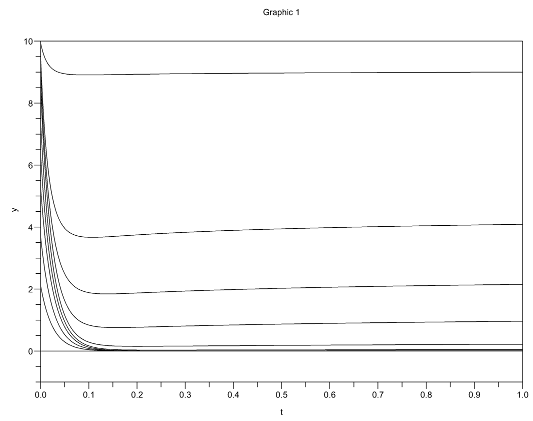


Figure 3: Time evolution of $CO_2$ concentration for different points on a tray

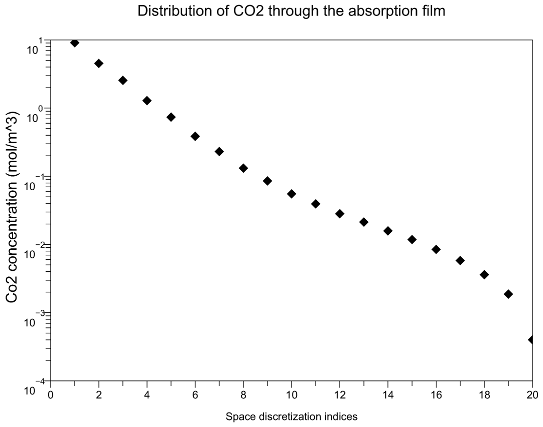


Figure 4: $CO_2$ concentrations at equilibrium state over a tray

same as that obtained with the previous models developed with gPROMs and Scilab. The main advantage of Modelica we observed is a huge reduction in the simulation time. In fact, the simulation of the new model does not take more than a few seconds.
Using Modelica in developing the model provides several advantages over classical methods used in IFP:

- The model of absorption column is stiff and needs higher order numerical methods for accurate simulation. Using Modelica for time simulation lets the user concentrating on the modeling the system rather than the way it can be simulated. The Modelica simulation tool (Here Scicos) is in charge of time simulation. It provides several numerical solvers for the time simulation of the model. In fact previous implementations of the absorption system at IFP required taking into account several simplifications to be able to simulate the model and get the result in a reasonable time.

- The developed model can be used as a component in a library to be used in other models. It can be used to compute the equilibrium state as well as a time simulation. It can also be used for parameter sizing and parameter estimation.

- Modelica is a standard modeling language allowing the model to be simulated with any Modelica simulator which gives some sort of durability to the model.

## CONCLUSION

In this paper, we have used the Modelica language to model the process of $CO_2$ The absorption in Methyldiethanolamine using model including partial differential equations. In order to reduce the number of equations resulting from conversion of the PDE into an ODE, we have applied the orthogonal collocation method. The results of the simulations demonstrates several advantages of Modelica and orthogonal collocation method in modeling the absorption column over standard methods used at IFP.

## REFERENCES

[1] S. L. Campbell, J.P. Chancelier, R. Nikoukhah, Modeling and simulation in Scilab/Scicos, Springer Verlag publishing, 2005.

[2] Levon Saldamli, Peter Fritzson, Bernhard Bachmann, Extending Modelica for partial Differential Equations, Modelica conference, 2008.

[3] Arthur Kohl, Richard Nielsen, Gas Purification, Fifth Edition, 1997

[4] John Polasek, and Jerry Bullin, Selecting Amines for Sweetening Units, Gas Processors Association Regional Meeting, Sept. 1994.

[5] Lars Erik Øi, Aspen HYSYS simulation of CO2 removal by amine absorption from a gas based power plant. Proceedings, SIMS 2007, Gothenburg, Sweden.

[6] Greer et al., A dynamic model for the de-absorption of carbon dioxide from Monoethanolamine solution, SIMS 2008, Oslo, Norway.

[7] J.M. LE Lann, A. Sargousse1, P. Sere Peyrigain, X. Joulia, Dynamic Simulation Of Partial Differential Algebraic Systems: Application to some Chemical Engineering problems, 3rd ICL Joint Conference; TOULOUSE,France, 1998

[8] M. P. Remelhe, Jochen Till,'Modeling of Dynamic Systems',2005

[9] J. Villadsen, M. Michelsen,'Solution of Differential Equation Models by Polynominal Approximation', Prentice-Hall pub., Englewood Cliffs, New Jersey, 1982

[10] R. G. Rice, D. D. Do, 'Applied Mathematics and Modeling for Chemical Engineers', Wiley pub., New York, 1995, chapters 8 and 12

[11] Y. S. Cho and B. Joseph, 'Reduced-order Steady-state and Dynamic Models for Separation Processes, Part I. Development of the Model Reduction Procedure', AIChEJ, 1983, vol 29, number 2, pp 261-269

[12] B. Carnahan, H. A. Luther,J. O. Wilkes, Applied Numerical Methods, John Wiley and Sons pub, New York, 1969, chapter 8 and 12

[13] R.S. Huss, 'Collocation Methods for Flexible distillation Design', Ph.D. Thesis, Dept. of Chem. Eng, Carnegie Mellon University, 1995

[14] H.M. Kvamsdala, J.P. Jakobsena, K.A. Hoffb, 'Dynamic modeling and simulation of a CO2 absorber column for post-combustion CO2 capture', Chem Eng and Proc: Process Intensification Volume 48, Issue 1, Jan 2009, pp 135-144.

[15] L. R. Petzold, "A Description of DASSL : A Differential/Algebraic System Solver", Proceedings of the 10th IMACS World Congress, Montreal,1982.

[16] D. Wallace,'Capture and Storage of CO2 what needs to be done?', November 2000, Available from http://www.earthscape.org/r1/iea05/

[17] Versteeg G. F., Van Dijck L. A. J., Van Swaaij W. P. M., 'On the kinetics between CO2 and alkanolamines both in aqueous and non-aqueous solutions, an overview', Chem. Eng. Comm. 144, 113-158, 1966

[18] Danckwerts, P.V., Sharma, M.M. 'The absorption of Carbon Dioxide into solutions of alkalis and amines', The Chemical Engineer, pp 244-280, 1966

[19] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward, "SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers," ACM Transactions on Mathematical Software, 31(3), pp. 363-396, 2005.

[20] A. C. Hindmarsh, "The PVODE and IDA Algorithms," LLNL technical report UCRL-ID-141558, December 2000.

[21] Najafi M., Benjelloun-Dabaghi Z., 'Modeling and Simulation of CO2 Absorber Column in Modelica',ESM2009 Conf., UK, 2009.

# EXPERIENCES OF USING THE PEPA PERFORMANCE MODELLING TOOLS WITH A NON-REPUDIATION PROTOCOL

Yishi Zhao
Nigel Thomas

School of Computing Science
Newcastle University, UK
Email: {Yishi.Zhao | Nigel.Thomas}@ncl.ac.uk

## ABSTRACT

In this paper, we described a experience of using PEPA eclipse plug-in tool to specify a functional-equivalent representation of a non-repudiation protocol. The model is specified using Markovian process algebra PEPA. The basic model suffers from the well known state space explosion problem when tackled using *Continues Time Markov Chain* analysis. In order to modelling in a scalable way, *functional rates* has been adopted to avoid a unintended system behaviour. The *functional rates* have been specified in a *CMDL* (Chemical Model Definition Language) format which equivalently generated from the PEPA model by *PEPA eclipse plug-in*. This representation has been converted back to PEPA expression, and analyzed numerically.

**KEYWORDS**
PEPA, PEPA eclipse plug-in, Functional Rates, Non-repudiation

## INTRODUCTION

Functional rates have been utilized to eliminate functional equivalent PEPA components to avoid state space explosion, [1]. This kind of specification is currently only supported by IPC (International PEPA Compiler).

In this paper, we demonstrate an alternative way of solving PEPA model with *functional rates* using the PEPA eclipse plug-in. This tool cannot directly analyze a PEPA model with functional rates, but an equivalent *CMDL* format can be generated that can be solved by fluid flow analysis (based on ODE) or stochastic simulation directly with the tool. We apply this method to a non-repudiation protocol, in which the service behaviours are not able to be defined in PEPA in a scalable way without *functional rates*.

The purpose of performance modelling security protocols is investigating the trade-off between security and performance. It is clear that in order to add more functionality to a system that more execution time is required. However, in the case of security, the benefit accrued from any additional overhead is not easy to quantify and so it is very hard for the performance engineer to argue that a particular performance target should take precedence over a security goal. There have been efforts made by both the security and performance communities to address aspects of this problem [2, 3, 4]. A *Key Distribution Centre* (key exchange protocol) has been studied in our previous work, which shows the possibility of modelling by the stochastic process algebra PEPA and analysis by several alternative techniques [5, 6, 7].

The remainder of this paper is organised as follows. In the next section we introduce the non-repudiation protocol to be modelled. This is followed by a brief review of the stochastic process algebra PEPA and PEPA eclipse plug-in. We then introduce the process of composing a PEPA model with *functional rates* and generating the equivalent CDML format model, followed by numerical results. Finally some conclusions are drawn and areas of further work described.

## NON-REPUDIATION PROTOCOL

A non-repudiation service will prevent either of the principals involved from denying the contract after the agreement. The protocol depicted here were proposed by Zhou and Gollmann [8] and use a non-repudiation server, known as a *Trusted Third Party* (TTP).

- L: a unique label chosen by $TTP$ to identify the message $M$
- $T_s$ : the time that $TTP$ received $A$'s submission
- $T_d$ : the time that $TTP$ delivered and available to $B$
- $NRO = sS_A(f_{NRO}, TTP, B, M)$ : non-repudiation of origin for $M$
- $NRS = sS_D(f_{NRS}, A, B, T_s, L, NRO)$ : non-repudiation of submission of $M$
- $NRR = sS_B(f_{NRR}, TTP, A, L, NRO)$ : non-repudiation of receiving a message labelled $L$
- $NRD = sS_D(f_{NRD}, A, B, T_d, L, NRR)$ : non-repudiation of delivery of $M$

In this protocol, $A$ sends the plaintext ($M$) and a non-repudiation origin ($NRO$) to the trusted third part ($TTP$), and then fetches the time of receiving ($T_s$) and non-repudiation of submission ($NRS$) from a public area, after $TTP$ has published this information. The $TTP$ tells $B$ it received $M$ from $A$ by sending the $NRO$. $B$ generates a non-repudiation of receiving for TTP following. Finally, $B$ and $A$ can fetch $M$ and the time of delivery ($T_d$), with other non-repudiation evidence, from the public area, after the $TTP$ has published.

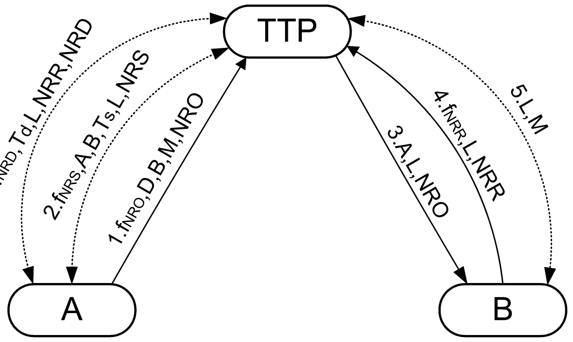

TTP
A
B
6.f<sub>NRD</sub>,Td,L,NRR,NRD
2.f<sub>NRS</sub>,A,B,Ts,L,NRS
1.f<sub>NRO</sub>,D,B,M,NRO
3.A,L,NRO
4.f<sub>NRR</sub>,L,NRR
5.L,M

Figure 1: Non-repudiation protocol invented by Zhou&Gollmann

$$\begin{array}{rll}
(request) & 1.A \rightarrow TTP: & f_{NRO}, TTP, B, M, NRO \\
(publish1\& & & \\
getByA1) & 2.A \leftrightarrow TTP: & f_{NRS}, A, B, T_s, L, NRS \\
(sendB) & 3.TTP \rightarrow B: & A, L, NRO \\
(sendTTP) & 4.B \rightarrow TTP: & f_{NRR}, L, NRR \\
(publish2\& & & \\
getByB) & 5.B \leftrightarrow TTP: & L, M \\
(publish2\& & & \\
getByA2) & 6.A \leftrightarrow TTP: & f_{NRD}, T_d, L, NRR, NRD
\end{array}$$

## PEPA AND PEPA ECLIPSE PLUG-IN

A formal presentation of PEPA is given in [9], in this section a brief informal summary is presented. PEPA, being a Markovian Process Algebra, only supports actions that occur with rates that are negative exponentially distributed. Specifications written in PEPA represent Markov processes and can be mapped to a continuous time Markov chain (CTMC). Systems are specified in PEPA in terms of *activities* and *components*. An activity $(\alpha, r)$ is described by the type of the activity, $\alpha$, and the rate of the associated negative exponential distribution, r. This rate may be any positive real number, or given as unspecified using the symbol $\top$.

The syntax for describing components is given as:

$$P ::= (\alpha, r).P|P + Q|P/L|P \bowtie_{\mathcal{L}} Q|A$$

The component $(\alpha, r).P$ performs the activity of type a at rate r and then behaves like $P$. The component $P + Q$ behaves either like $P$ or like $Q$, the resultant behaviour being given by the first activity to complete.

The component $P/L$ behaves exactly like $P$ except that the activities in the set $L$ are concealed, their type is not visible and instead appears as the unknown type $\tau$.

Concurrent components can be synchronised, $P \bowtie_{\mathcal{L}} Q$, such that activities in the cooperation set $L$ involve the participation of both components. In PEPA the shared activity occurs at the slowest of the rates of the participants and if a rate is unspecified in a component, the component is passive with respect to the activities of that type. $A \overset{def}{=} P$ gives the constant $A$ the behaviour of the component $P$. The shorthand $P||Q$ is used to denote synchronisation over no actions, i.e. $P \bowtie_{\emptyset} Q$. We employ some further shorthand that has been commonly used in the study of large parallel systems. We denote $A[N]$ to mean that there are $N$ instances of $A$ in parallel, i.e. $A|| \ldots ||A$, but we are not concerned with the state of each individual component, rather the number of components in each state. As such, when using this representation, we would not distinguish between $A||A'$ and $A'||A$.

In this paper we consider only models which are cyclic, that is, every derivative of components $P$ and $Q$ are reachable in the model description $P \bowtie_{\mathcal{L}} Q$. Necessary conditions for a cyclic model may be defined on the component and model definitions without recourse to the entire state space of the model.

The *PEPA eclipse plug-in* is developed by researchers based in Edinburgh University [10, 11]. The tool can be used to conduct steady state and transient analysis of PEPA models through solving the CTMC, stochastic simulation and fluid flow analysis (based on ODEs). PEPA models with *functional rates* that are used in this paper are not directly supported by current version of *PEPA eclipse plug-in*, but an equivalent CMDL file (Chemical Model Definition Language) can be generated in which functional rates can be specified and analyzed.

## MODEL CONSTRUCTION

### Initial Model

We intuitively begin by define the behaviour of a pair of principals as:

$$\begin{aligned}
A0 &\overset{def}{=} (request, r_{t1}).A1 \\
A1 &\overset{def}{=} (publish1, r_{p1}).A2 \\
A2 &\overset{def}{=} (getByA1, r_{ga1}).A3 \\
A3 &\overset{def}{=} (sendB, r_b).A4 \\
A4 &\overset{def}{=} (publish2, r_{p2}).A5
\end{aligned}$$

$$\begin{aligned}
A5 &\stackrel{def}{=} (getByA2, r_{ga2}).A6\\
A6 &\stackrel{def}{=} (work, r_w).A0\\
\\
B0 &\stackrel{def}{=} (sendB, r_b).B1\\
B1 &\stackrel{def}{=} (sendTTP, r_{t2}).B2\\
B2 &\stackrel{def}{=} (publsih2, r_{p2}).B3\\
B3 &\stackrel{def}{=} (getByB, r_{gb}).B4\\
B4 &\stackrel{def}{=} (work, r_w).B0\\
\\
TTP &\stackrel{def}{=} (publish1, r_{p1}).TTP\\
&\quad +(publish2, r_{p2}).TTP\\
&\quad +(sendB, r_b).TTP\\
\\
System &\stackrel{def}{=} TTP[K] \underset{\mathcal{L}}{\bowtie} (A0||B0)[N]
\end{aligned}$$

Where, $\mathcal{L} = \{publish1, publish2, sendB\}$.

In order simplify the model specification and analysis, we combine $A$ and $B$ into a new component called $AB$, using a process referred to as *partial evaluation* [12]. Following description can be obtained as a strong equivalent representation of above one with $N$ pairs of principals.

$$\begin{aligned}
AB0 &\stackrel{def}{=} (request, r_{t1}).AB1\\
AB1 &\stackrel{def}{=} (publish1, r_{p1}).AB2\\
AB2 &\stackrel{def}{=} (getByA1, r_{ga1}).AB3\\
AB3 &\stackrel{def}{=} (sendB, r_b).AB4\\
AB4 &\stackrel{def}{=} (sendTTP, r_t2).AB5\\
AB5 &\stackrel{def}{=} (publish2, r_{p2}).AB6\\
AB6 &\stackrel{def}{=} (getByA2, r_{ga2}).AB7\\
&\quad +(getByB, r_{gb}).AB8\\
AB7 &\stackrel{def}{=} (getByB, r_{gb}).AB9\\
AB8 &\stackrel{def}{=} (getByA2, r_{ga2}).AB9\\
AB9 &\stackrel{def}{=} (work, r_w).AB0;\\
TTP &\stackrel{def}{=} (publish1, r_{p1}).TTP\\
&\quad +(publish2, r_{p2}).TTP\\
&\quad +(sendB, r_b).TTP\\
System &\stackrel{def}{=} TTP[k] \underset{publsih1, publish2, sendB}{\bowtie} AB0[N]
\end{aligned}$$

$AB_0$ to $AB_9$ in the above PEPA model denote the different behaviours of the $AB$ component, and its evolution along the sequence of prescribed actions in the protocol. The choice from $AB_6$ to $AB_7$ and $AB_8$ means step 5 and step 6 in the protocol can happen in any order. The $work$ action is used to define that $B$ can do something with the plaintext (M) that was sent by $A$ after he has obtained it, before returning to the state $AB_0$ to make a new request again, which forms a working cycle to investigate the steady state. Here, we assume multiple TTP servers specified in the model are able to share a common memory.

In this TTP component representation, this naive model gives rise to a race between $publish1$, $publish2$ and $sendB$ in PEPA, which does not capture the intended behaviour of actual system. To avoid this unwanted race, a reasonable solution is adjusting the three rates associated with it, that should depend on the proportion each job that request service of $publish1$, $publish2$ and $sendB$ respectively. In other words, the rates are dependent on the current state of system using so-called *functional rates*. The procedure of specifying *functional rates* is illustrated in next subsection.

### Functional Rates Specification

In this subsection we describe a *CDML (Chemical Model Definition Language)* model generated from above PEPA model by the PEPA eclipse plug-in. This specification contains rate functions and is able to be analyzed by the fluid flow approach (based on ODEs) or stochastic simulation, supported by the tool. The following *CMDL* model is generated by eclipse plug-in and modified with *functional rates*.

$$\begin{array}{ll}
//Rates & //Population\ sizes\\
rb = 1.0; & AB0 = N;\\
rga1 = 1.0; & AB1 = 0;\\
rga2 = 1.0; & AB2 = 0;\\
rgb = 1.0; & AB3 = 0;\\
rp1 = 1.0; & AB4 = 0;\\
rp2 = 1.0; & AB5 = 0;\\
rt1 = 1.0; & AB6 = 0;\\
rt2 = 1.0; & AB7 = 0;\\
rw = 0.01; & AB8 = 0;\\
 & AB9 = 0;\\
 & TTP = K;
\end{array}$$

$$\begin{array}{l}
//Reactions\\
getByA1, AB2 \rightarrow AB3, rga1;\\
getByA21, AB6 \rightarrow AB7, rga2;\\
getByA22, AB8 \rightarrow AB9, rga2;\\
getByB1, AB6 \rightarrow AB8, rgb;\\
getByB2, AB7 \rightarrow AB9, rgb;\\
publish1, TTP + AB1 \rightarrow TTP + AB2, rx1;\\
publish2, TTP + AB5 \rightarrow TTP + AB6, rx2;\\
request, AB0 \rightarrow AB1, rt1;\\
sendB, TTP + AB3 \rightarrow TTP + AB4, rx3;\\
sendTTP, AB4 \rightarrow AB5, rt2;\\
work, AB9 \rightarrow AB0, rw;
\end{array}$$

Where,
$r_{x1} = [rp1 * AB1 * ((AB1 + AB3 + AB5)^{-1}) * min(TTP, AB1 + AB3 + AB5)]$
$r_{x2} = [rp2 * AB5 * ((AB1 + AB3 + AB5)^{-1}) *$

$min(TTP, AB1 + AB3 + AB5)]$
$r_{x3} = [rb * AB3 * ((AB1 + AB3 + AB5)^{-1}) * min(TTP, AB1 + AB3 + AB5)]$

This *CMDL* format of the model is composed of *Rates*, *Population sizes* and *Reactions* parts. The *Rates* section is exactly the same as that specified in the PEPA model. The *Population sizes* contains the initial population of all derivatives and components. In our scenario, there are $N$ client pairs, which haven't started any behaviours at the initial stage, represented by $AB_0 = N$, other derivatives have no population. $K$ is the population of TTP all the time as no derivatives associated with it. The most important and main section of *CMDL* definition is $Reactions$, in which system behaviours defined as all actions name, individual state transitions and their rates.

In the *CMDL* model, we specify the functional rates for each cooperation under action $publish1$, $publish2$ and $sendB$ by $r_{x1}$, $r_{x2}$ and $r_{x3}$, respectively, instead of $r_{p1}$, $r_{p2}$ and $r_b$. Each of these functions describes the actual service rate if there is one job in the system($r_{p1}$, $r_{p2}$ or $r_b$), or as a proportion of the number of waiting jobs of each type ($ABi * ((AB1 + AB3 + AB5)^{-1}), i = 1, 3, 5$) and the times of service ($min(TTP, AB1 + AB3 + AB5)$), which allocates each service with respect to its job type to eliminates the potential race. Although the PEPA model with function rates cannot be recognised by *PEPA eclipse plug-in*, it is still necessary to write it down as a formal specification of the protocol:

$$\begin{array}{rcl}
AB_0 & \overset{def}{=} & (request, r_{t1}).AB_1 \\
AB_1 & \overset{def}{=} & (publish1, r_{x1}).AB_2 \\
AB_2 & \overset{def}{=} & (getByA1, r_{ga1}).AB_3 \\
AB_3 & \overset{def}{=} & (sendB, r_{x3}).AB_4 \\
AB_4 & \overset{def}{=} & (sendTTP, r_{t2}).AB_5 \\
AB_5 & \overset{def}{=} & (publish2, r_{x2}).AB_6 \\
AB_6 & \overset{def}{=} & (getByB, r_{gb}).AB_7 \\
 & & +(getByA2, r_{ga2}).AB_8 \\
AB_7 & \overset{def}{=} & (getByA2, r_{ga2}).AB_9 \\
AB_8 & \overset{def}{=} & (getByB, r_{gb}).AB_9 \\
AB_9 & \overset{def}{=} & (work, r_w).AB_0 \\
TTP & \overset{def}{=} & (publish1, r_{x1}).TTP \\
 & & +(publish2, r_{x2}).TTP \\
 & & +(sendB, r_{x3}).TTP \\
System & \overset{def}{=} & TTP[K] \underset{publish1, publish2, sendB}{\bowtie} AB_0[N]
\end{array}$$

Where,
$r_{x1} = r_{p1} \frac{AB_1(t)}{AB_1(t)+AB_3(t)+AB_5(t)} min(AB_1(t) + AB_3(t) + AB_5(t), TTP(t))$,
$r_{x2} = r_{p2} \frac{AB_5(t)}{AB_1(t)+AB_3(t)+AB_5(t)} min(AB_1(t) + AB_3(t) + AB_5(t), TTP(t))$.
$r_{x3} = r_b \frac{AB_3(t)}{AB_1(t)+AB_3(t)+AB_5(t)} min(AB_1(t) + AB_3(t) + AB_5(t), TTP(t))$.

## NUMERICAL RESULTS

In our previous work [7], an assumption of the same action name and the same rates has been made for $publish1$, $publish2$ and $sendB$. With functional rates a more general scenario can be investigated to observe any differences between these three types of TTP service.

Figure 2 shows the average queue length varied with number of customer involved in this non-repudiation system solved by ODE solution supported by the tool. The ODE solution is an approximation which is very accurate in the extremes ($N = 1$ or large $N$) but much less accurate at the point at which the angle of the plot changes (here around $N = 14$ or $N = 16$). The point of maximum error can be precisely predicted, following our previous research [6]. If an accurate solution is required at these points then stochastic simulation could be employed within the PEPA eclipse plug-in.

Obviously the queue length increases as more client pairs join the system for both cases. In the case of $r_{p1} = 0.5$ and $r_{p2} = 0.2$, number of waiting jobs is always larger than the other case, as the average service rate is lower. In addition, the queue length of this set of parameters increases faster, this because the slower server is proportionally more heavily loaded as demand increases.

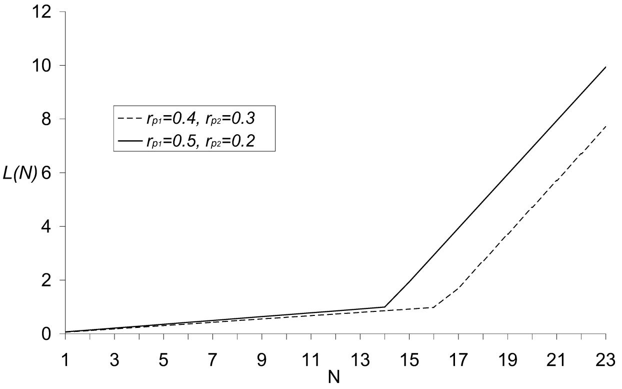


Figure 2: Average queue length varied with population size calculated by the ODE, $r_b = r_{t1} = r_{ga1} = r_b = r_{t2} = r_{gb} = r_{ga2} = 1, r_w = 0.01, K = 1$

Following the random observer principle of queueing theory (see [13] for example), the average response time can be calculated. If the random observer sees a free server, then the average response time will be the average service time, i.e. there is no queueing. However, if the random observer sees all the servers busy, then the average response time will be the average service time plus the time it takes for one server to become available (including scheduling the other jobs waiting ahead of the random observer).

$$
\begin{aligned}
W(N) &= \frac{1}{r_p}\ ,\ L(N-1)+1 \leq K \\
W(N) &= \frac{1}{r_p} + \frac{L(N-1)+1-K}{Kr_p} \\
&= \frac{L(N-1)+1}{Kr_p}\ ,\ L(N-1)+1 > K
\end{aligned}
$$

The above equations have been adopted to calculate average response time varied with system capacity by individual service behaviours in Figure 3. Here $W(1)$, $W(3)$ and $W(5)$ denotes the response times for the three responding actions by the TTP in the protocol, with the rates $r_{p1}$,$r_b$ and $r_{p2}$ respectively. These are equivalent to the derivatives $AB_1$, $AB_3$ and $AB_5$ in the PEPA model. Clearly, the average response time for the first job type is slightly larger than third one ($AB_5$) and smaller than for the second job type ($AB_3$), because of the ratio between response time and responding rate. However, the average response time of all three job types grows at the same rate. The reason is obviously that the time for processing all the requests already within the queue is the same, only the time to process the arriving request differs. Thus, the difference between the response times of these three response actions is a constant.

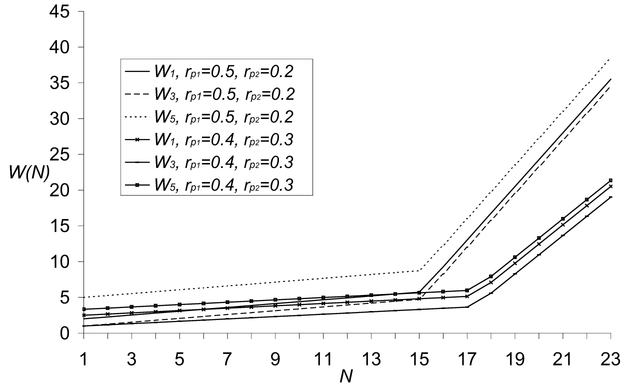


Figure 3: Average response time varied with population size calculated by the ODE, $r_b = r_{t1} = r_{ga1} = r_b = r_{t2} = r_{gb} = r_{ga2} = 1, r_w = 0.01, K = 1$

It is also interesting to note that the differences that occur as we alter the rate of the second and third response action. This difference between the two sets of curves is quite significant, far more so than we might naively expect. The initial stage ($N = 1 \sim 6$) of the average response time of the first type of jobs ($W(1)$) becomes larger as response rate decreases. Nevertheless, all three job types tend to respond quicker than the first set as $N$ increases, because the average service time ($1/rp1 + 1/rb + 1/rp2$) is decreased, and the proportion of this type of request waiting at the $TTP$ is smaller.

Multiple servers can be analyzed, as illustrated in Figure 4. Here, $L(1)$, $L(3)$ and $L(5)$ denote the queuing lengths for the three responding actions by the TTP in the protocol, corresponding to $AB_1$, $AB_3$ and $AB_5$ in the PEPA model. In each set of curves, a larger service rate results in a smaller number of waiting customers. Generally, there are fewer jobs waiting if more servers are being provided (obviously). Nevertheless, the number of the first type of waiting jobs (L1) with four $TTP$ servers catches number of second type jobs with two $TTP$ servers when $N = 145$, as they are the slowest and fastest one in each set respectively.

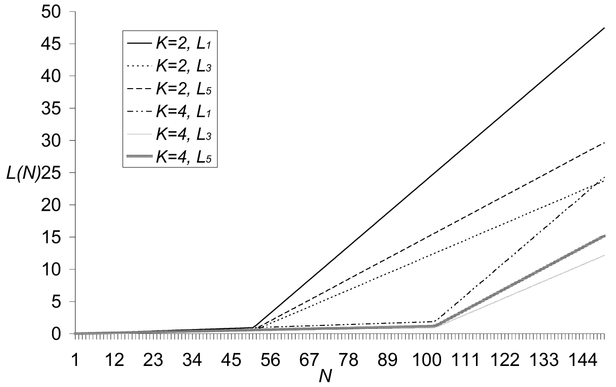


Figure 4: Average queue length varied with population size with different number of servers calculated by the ODE, $r_b = r_{t1} = r_{ga1} = r_b = r_{t2} = r_{gb} = r_{ga2} = 1, r_{p1} = 0.5, rp2 = 0.8, r_w = 0.01$

## CONCLUSION AND FUTURE WORK

In this paper, we have showed how functional rates can deal with the unintended behaviours in the system, and introduced a novel approach to specify and analysis a PEPA model of a non-repudiation protocol with functional rates using PEPA eclipse plug-in. Although this tool cannot directly solve PEPA model with functional rates, a *CMDL* format model can be generated which can use functional rates and is supported by the tool with time series analysis. The PEPA eclipse plug-in includes ODE analysis and stochastic simulation, which are very scalable approaches, and applicable for a large class of models. Some numerical results have been presented which benefit from the functional rates specification.

There is still a limitation, in that the average response time calculation with different service rates at the $TTP$ with multiple servers cannot be calculated exactly. In this case, in order to obtain the response time, the time it takes for one $TTP$ server to become available should be calculated first. However, in FCFS queueing this requires us to know the queued order of the requests, which is clearly infeasible. We can only obtain the response time for three responding rates when there is a single $TTP$ server. Thus, the waiting time for an arriving request is the time for a single $TTP$ server to respond to all the requests in the queue, which does not require any knowledge about the order in which requests are queued. This remains issues of further investigation.

The work presented here forms part of an ongoing investigation into techniques for modelling and performance analysis of security protocols. The motivation for this work is the need to be able to investigate the trade-off that often exists between providing a secure environment and one that meets its temporal requirements. In the continuation of this investigation we will consider further protocols with more complex behaviour, e.g. multiple authentication parties and broadcast mechanisms.

## REFERENCES

[1] J. Hillston and L. Kloul, A Functional Equivalent Component Based Simplification Technique for PEPA Models, in: *3rd European Performance Engineering Workshop*, pp.16-30, LNCS 4054, Springer-Verlag, 2006.

[2] S. Dick and N. Thomas, *Performance analysis of PGP*, in: F. Ball (ed.) *Proceedings of 22nd UK Performance Engineering Workshop*, Bournemouth University, 2006.

[3] W. Freeman and E. Miller, An Experimental Analysis of Cryptographic Overhead in Performance-critical Systems, *Proceedings of the 7th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, IEEE Computer Society, 1999.

[4] C. Lamprecht, A. van Moorsel, P. Tomlinson and N. Thomas, Investigating the efficiency of cryptographic algorithms in online transactions, *International Journal of Simulation: Systems, Science & Technology*, 7(2), pp 63-75, 2006.

[5] Y. Zhao and N. Thomas, Approximate solution of a PEPA model of a key distribution centre, in: *Performance Evaluation - Metrics, Models and Benchmarks: SPEC International Performance Evaluation Workshop*, pp. 44-57, LNCS 5119, Springer-Verlag, 2008.

[6] N. Thomas and Y. Zhao, Fluid flow analysis of a model of a secure key distribution centre, in: *Proceedings of the 24th UK Performance Engineering Workshop*, Imperial College London, 2008.

[7] N. Thomas and Y. Zhao, Mean value analysis for a class of PEPA models, in: *6th European Performance Engineering Workshop*, pp.59-72, LNCS 5652, Springer-Verlag, 2009.

[8] J. Zhou and D. Gollmann, Observation on Non-repudiation, in: *Advances in Cryptology-ASIACRYPT'96*, pp.133-144, LNCS 1163/1996, Springer-Verlag, 1996.

[9] J. Hillston, A Compositional Approach to Performance Modelling, pp.141, Cambridge University Press, 1996.

[10] M. Tribastone, The PEPA Plug-in Project, in: *Proceedings of the 4th International Conference on the Quantitative Evaluation of SysTems (QEST)*, pages 53-54, IEEE, 2007.

[11] http://eclipse.org.

[12] A. Clark, A. Duguid, S. Gilmore and M. Tribastone, Partial Evaluation of PEPA Models for Fluid-Flow Analysis, in: *Computer Performance Evaluation: 5th European Performance Engineering Workshop*, LNCS 5261, Springer-Verlag, 2008.

[13] I. Mitrani, Probabilistic Modelling, Cambridge University Press, 1998.

## BIOGRAPHY

**Yishi Zhao** is currently completing his Ph.D. in the School of Computing Science at Newcastle University. His area of investigation is the performance modelling and analysis of security protocols using stochastic process algebra. Yishi was awarded an M.Sc. in Computing Science with Distinction from Newcastle University in 2006.

**Nigel Thomas** is a Reader in the School of Computing Science at Newcastle University. He joined the Newcastle University in January 2004 from the University of Durham, where he had been a lecturer since 1998. His research interests lie in performance modelling, in particular Markov modelling through queueing theory and stochastic process algebra. Nigel was awarded a Ph.D. in 1997 and an M.Sc. in 1991, both from the Newcastle University.

# THE CO-SPACE PROJECT – A FRAMEWORK AND SET OF TOOLS FOR THE BUILDING AND VISUALIZATION OF SENSOR, TELEMETRY & WEB-INTEGRATED SIMULATIONS WITHIN WONDERLAND WORLDS

Chong-Wee Simon See, Chee-Kian Melvin Koh,
Che-Wing Cheung
Asia Pacific Science and Technology Center,
Sun Microsystems Pte Ltd.
1 Magazine Road, #07-01/13, Central Mall,
Singapore 059567
Email:{simon.see | melvin.koh | che.wing}@sun.com

Douglas Finnigan, Joon Jew Liow

Temasek Informatics & IT School
Temasek Polytechnic
21 Tampines Avenue 1
Singapore 529757
Email: {douglas | joonyew}@tp.edu.sg

**KEYWORDS**
Sensors, Sun SPOT, Sun Wonderland, virtual worlds in simulation, visualisation

## ABSTRACT

Despite the increasing use of virtual world technologies for serious applications, there is a lack of tools for the building and visualization of immersive simulations. The present paper describes the aims of the Co-Space project, and outlines a framework and set of tools that facilitate such development.

## INTRODUCTION

The increasing popularity of virtual worlds, coupled with the availability of cheap computer systems capable of supporting complex, distributed 3D applications, has encouraged research into more serious uses of virtual world technologies. Recent projects include the study of consumer behavior in a virtual world (Wetsch et al. 2008), the use of mixed reality teaching and learning (Callaghan et al. 2008), and a Canadian border simulation for the training of customs officers (Hudson et al. 2009).

One of the most promising uses of virtual worlds is the collaborative monitoring of, and interaction with, the real world. On the monitoring side, the use of sensors and dynamic web feeds allow data to be input and accessed in a way that groups of people can easily visualize, manipulate, and analyze. For students especially, the immersive nature of virtual world based simulations is a powerful example of how "new technologies give rise to many more opportunities for simulation-based learning, in which a person is placed in a scenario or situation and is directly responsible for the changes that occur as a result of his or her decisions" (Chodos et al. 2009). For example, rather than simply clicking a 2D button to start a factory floor simulation, the user would walk up to a button located correctly with regard to a physical factory floor, and press the button; indeed, this would feel more like performing a real task than running a simulation. Fujimoto makes a comparison between virtual environments and analytic simulations (Fujimoto 2000). He comments that virtual environments are well suited to the testing of physical components in situations where live testing would be impractical or dangerous, such as a missile defense system. The embedding of such devices allows direct human participation in the simulation.

While it is possible to implement simulation specific functionality in popular virtual world platforms such as Second Life (Lang and Kobilnyk 2009) this is often constrained by architectural and business limitations imposed by the system; for example, Second Life has a restrictive art pipeline (van Nederveen 2007).

Project Wonderland, part of the Collaborative Environments project at Sun Microsystems, is an open source toolkit for building virtual worlds (Yankelovich and Kaplan 2008). Written entirely in Java, Wonderland builds on other open-source projects, including Project Darkstar for scalable distributed gaming services, jMonkeyEngine for professional-level 3D rendering, jVoiceBridge for voice communication and in-world audio, and also includes the sharing of X11 applications between clients. A key design factor is openness and extensibility, whereby developers can extend Wonderland's functionality easily without having to modify the core source code. Extensions include the creation of new worlds, the addition of new art workflows (including in-world building), and the development of new features and behaviors for Wonderland objects and avatars. The open-source nature of Wonderland allows developers to make use of other open-source projects to extend the capabilities of the server.

This paper describes the Co-Space project, sponsored by the Media Development Authority of Singapore. One of the key objectives of Co-Space is to explore the potential for Wonderland to interface with physical devices, thus enabling a seamless integration between the real and virtual worlds. The Co-Space project intends to develop sensor input and simulation frameworks that facilitate the implementation of such ideas. Though the project is only in its initial stages, the following sections describe the aims of the Co-Space project, and the work done so far.

## THE CO-SPACE PROJECT

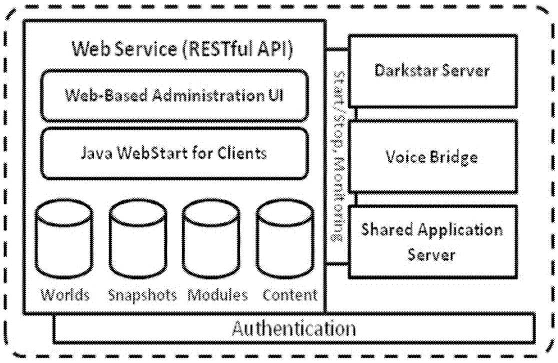
The Co-Space Project is supported by the multi-agency Interactive Digital Media R&D Programme Office (IDMPO) hosted by the Media Development Authority of Singapore. IDMPO was established in 2006 under the mandate of the National Research Foundation to deepen Singapore's research capabilities in interactive digital media (IDM), fuel innovation and shape the future of media.

Co-Space consists of two sub-projects, one focused on interfacing Wonderland with physical sensors and real-world data feeds (Sensor Project), the other examining how this data can be visualized and used in-world (Simulation & Visualization Project). More formally, the Sensor Project is developing a Wonderland Framework for Sensor, Telemetry & Web Integration, enabling Wonderland to allow a consistent integration of sensors and real-world telemetry data. Sensors may be next generation gadgets such as Sun SPOTs (Small Programmable Object Technology devices) or mobile phones with motion sensor or Bluetooth™, while telemetry data may be from medical monitoring devices such as heart monitors. In addition, there is a tremendous amount of 2D and 3D data available via web services.

We plan to enable both sensor and web service data to be integrated into Wonderland worlds. Our goal is to make it easy for people to create virtual worlds that are highly dynamic and constantly up-to-date. We have implemented two scenarios as initial proofs of concept: the input, processing and visualization of Yahoo! Weather RSS data; and the interfacing of Wonderland with Sun SPOT data. Before discussing implementation details, we outline the features of Wonderland that make it so suitable for such work.

### The Wonderland Architecture

One of the most powerful innovations of Wonderland's design is its modular architecture, in which functionality is extended by the creation of modules that are uploaded to the Wonderland server via a simple web-based interface. Modules are simply wrappers for uploading and deploying art content, code and other services to the server. Once installed, the Wonderland server uses the module structure to unpack the various parts and distributes them accordingly. Modules are central to Wonderland, and much of the core software is packaged into modules. Writing new modules is straightforward, enabling developers to focus on core functionality rather than on how to interface with the Wonderland server. Wonderland includes pre-installed modules for avatar control and rendering, a Collada 3D file loader, a portal for transporting within and between worlds, and a module for sharing state information between clients. The Wonderland server architecture is illustrated in Figure 1.


Web Service (RESTful API)
Web-Based Administration UI
Java WebStart for Clients
Worlds
Snapshots
Modules
Content
Start/Stop, Monitoring
Darkstar Server
Voice Bridge
Shared Application Server
Authentication

Figures 1: The Wonderland Server Architecture

Visible objects are represented by cells, which can be thought of as 3D volumes, equivalent to nodes in a 3D scene graph. Cells may contain one or more components, each of which implements a particular functionality, or feature, of a cell. Components are not specific to cells, and different types of cells may contain the same types of components, where appropriate.

### Simulation and Visualization Project

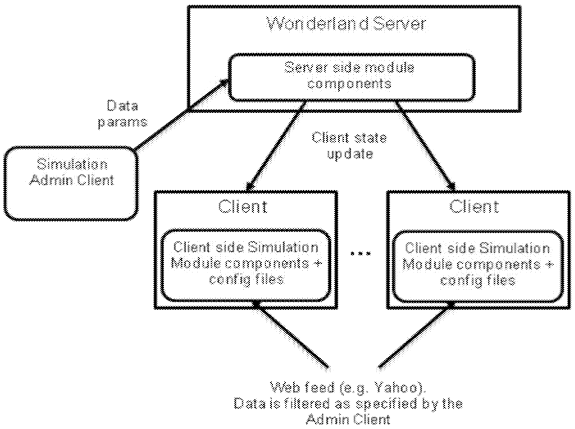
The Simulation and Visualization Project is focused on developing a Framework and Tool for Building Simulations within Wonderland Worlds. As mentioned above, an initial proof of concept is the input, processing and visualization of Yahoo! Weather RSS data. Initial setup is via a Simulation Admin Client, which is responsible for:

- configuring the visualization;
- configuring data access (e.g. URL, data selection, polling time);
- instantiating the Simulation Module;
- communicating with the Simulation Module.

The actual visualization is implemented as a Simulation Module, instantiated by the Simulation Admin Client on startup. One of the components of this Simulation Module is a renderer, responsible for instantiating and configuring a particular visualization. The Simulation Module is responsible for:

- receiving the simulation data from the Simulation Admin Client;
- creating the appropriate simulation and visualization objects;
- maintaining the state of these objects.

One advantage of this approach is that regular Wonderland clients do not need to install any additional modules, which is usually done via the client GUI. A disadvantage is that interaction with the simulation is not possible, though future work will explore the possibility of allowing real-time client modification of simulation parameters. The high-level architecture of the Simulation Admin Client and Simulation Module is illustrated in Figure 2.

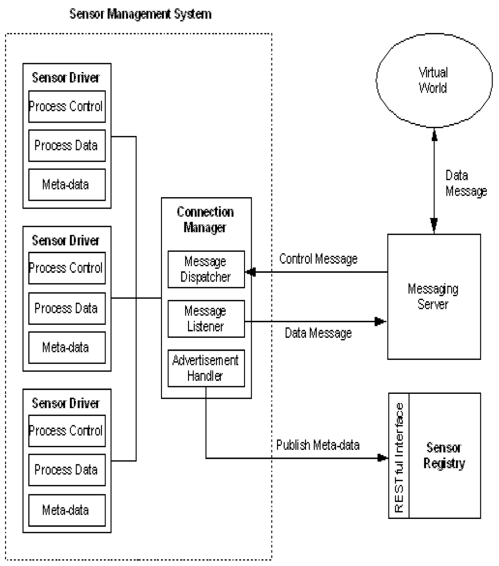

Wonderland Server
Server side module components
Data params
Client state update
Simulation Admin Client
Client
Client side Simulation Module components + config files
...
Client
Client side Simulation Module components + config files
Web feed (e.g. Yahoo).
Data is filtered as specified by the Admin Client

Figures 2: High-level architecture of the Simulation Admin Client and Simulation Module

The Simulation Admin Client is used to specify a particular URL for the weather data, and to configure which data values the visualization should use, depending on requirements (e.g. wind direction and temperature, but not pressure). The Simulation Admin Client instantiates the Simulation Module components on the server, which then downloads and configures the client side components for each client. Each client then accesses and creates a visualization for the specified data from the given URL, the actual visualization in this case implemented using a jMonkeyEngine particle system. The Yahoo weather feed includes a “condition code”, which specifies a particular type of weather, e.g. 9 represents drizzle, 16 represents snow. Casti discusses how the collection of such data represents the first step in building a numerical weather-forecasting model (Casti, J. 1997) . Particle system parameters for each type of weather code are stored in simple configuration files within the Simulation Module, which can be edited to create customized particle effects. In addition, the modular nature of Wonderland allows more than one type of cell renderer to be packaged on the client, and so different types of visualizations, such as weather charts, can be used.

**Sensor Project**

The Sensor Project is focused on developing a framework for interfacing with sensors and other real-world data streams. This framework will be used to interface with virtual worlds such as Wonderland for collaborative monitoring and interacting with the real-world. On the monitoring side, it allows the use of sensors to capture data for presentation in the virtual world. Conversely, virtual worlds can interact with the real world by sending control messages to actuators. The framework is designed to allow new sensors to be easily added. Three types of sensor devices will be integrated initially: Sun SPOTs, with sensors for temperature, light and motion; virtual sensors which extract measurement data from the web; and mobile sensor devices such as mobile phones or PDAs. The high-level architecture of the Sensor Framework is shown in Figure 3.



Figures 3: High-level architecture of the Sensor Framework

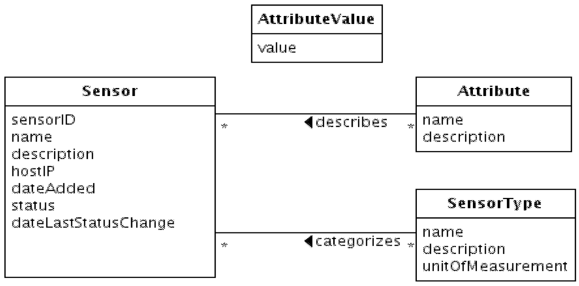
The Sensor Framework consists of three main systems – a Sensor Messaging System, a Sensor Registry and a Messaging Server.

Messaging Server

The Messaging Server is mainly responsible for routing the messages to their intended target. Currently, the messaging protocol used by the framework is the widely adopted Extensible Messaging and Presence Protocol (XMPP). Building on top of the XMPP protocol, we are planning to define our own set of protocols to support controlling of actuators, publish/subscribe and polling of sensor data. As a proof of concept, the OpenFire server is being used, which is open source under the GPL license.

**Sensor Messaging System**

The Sensor Messaging System (SMS) is the main workhorse of the framework. This system is responsible for collecting and dispatching measurements from the sensors. The SMS provides a sensor abstraction interface which allows easy integration of new sensors by developing new drivers which extend these interfaces for each sensor. As it is possible for a SMS to host more than one sensor, it maintains a separate processing thread for each driver. These drivers communicate with the sensors, collecting and processing the measurements which are then passed to the Connection Manager for dispatching to the Messaging Server.



Figures 4: UML class diagram of the domain model for the Sensor Registry

Sensor

Each Sensor has a unique ID, with a name and description for easy identification.  Each Sensor also stores the IP location of the hosting Sensor Management System, the Sensor’s online availability status, and a time-stamp indicating when this information was last updated. An example Sensor is a temperature sensor on a SUN SPOT device.

SensorType

The SensorType broadly categorizes Sensors.  A Sensor may belong to more than one SensorType.  A unit of measurement field allows differentiation between sensors of the same type but which use different units of operation. An example SensorType is “temperature”. Note that there may be two temperature SensorTypes, one measuring in degrees Fahrenheit, the other in Celsius.

Attribute

An Attribute gives extra information about a Sensor. Example Attributes are “Keyword” and “Location”. A Sensor may have more than one Attribute.

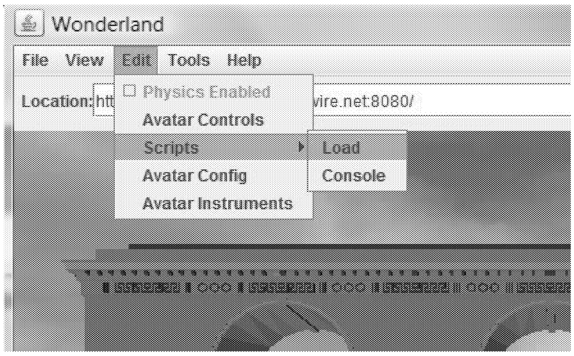
AttributeValue

A Sensor might have more than one value for any of its Attributes, e.g. a Sensor might have many values for its Keyword attribute.

The Sensor Registry exposes a RESTful web service to users. The RESTful web service can be accessed by a wide variety of clients, encouraging greater use of the Sensor Registry services and the Sensors it exposes. In addition to the Jersey implementation of JAX-RS (Java API for RESTful Web Services) used to implement the RESTful web services, JPA (Java Persistence API) and JAXB (Java Architecture for XML Binding) APIs were used to implement data persistence and XML data formatting.

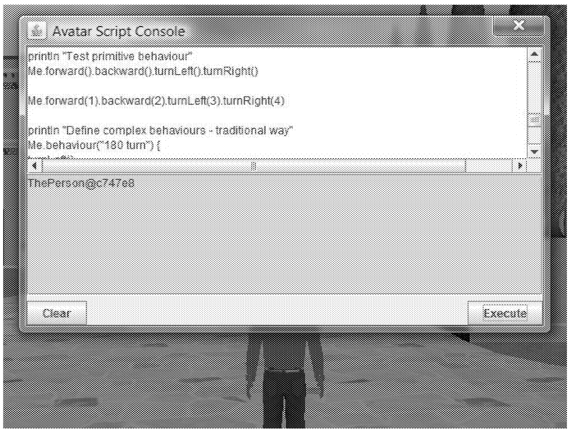
### Scripting

An additional goal of the Co-Space project is the implementation of scripting capabilities in Wonderland. Written in Java, Wonderland is able to make use of the JSR 223 standard scripting interface and API (Bosanac, D 2007). A language such as Groovy can be used to develop a domain specific language (DSL) to facilitate simulation development and control (Koenig, D 2007). A prototype, in which the user’s avatar is controlled by a simple Groovy-based DSL, has been developed. Figure 5 shows a menu item for loading and running a Groovy script file.



Figures 5: Avatar Scripting Menu

The avatar can also be controlled by a script typed into a script console (Figure 5). Examples of the avatar control DSL can be seen in Figure 6, where “Me” is a variable representing the avatar. Simple commands, such as forward(), backward(), turnLeft(), turnRight() are used, with or without parameters, to specify the avatar’s movement.



```
println "Test primitive behaviour"
Me.forward().backward().turnLeft().turnRight()

Me.forward(1).backward(2).turnLeft(3).turnRight(4)

println "Define complex behaviours - traditional way"
Me.behaviour("180 turn") {
```

```
ThePerson@c747e8
```

Figures 6: Avatar Scripting Console

## CONCLUSION

In a short space of time, the World Wide Web has become an integral part of our lives, allowing us to create new forms of interaction between distributed communities in fundamental areas such as commerce, education, social networking and entertainment. Yet what the Web lacks, and what virtual environments give, is the sense of immersion, the feeling that we exist inside and interact directly with the virtual space we inhabit. As virtual world technologies evolve, this sense of immersion will increase, and the ability to move seamlessly between the real and virtual worlds will become commonplace. The aim of the Co-Space project is to develop tools that enable the easy integration of real-world data into the virtual environment. The development of such frameworks and tools to integrate physical devices and in-world simulations, is a (real) step in this direction. Rather than being simply an example of real-world data integration, the project’s simulation and visualization toolkit will facilitate the easy creation of mixed reality environments across a wide range of domains.

## REFERENCES

Bosanac, Dejan. 2007. *Scripting in Java: Languages, Frameworks, and Patterns*, Addison-Wesley.

Callaghan, V; Gardner, M; Horan, B; Scott, J; Shen, L; and Wang, M. 2008. “A Mixed Reality Teaching And Learning Environment”, *Lecture Notes In Computer Science Vol. 6169, 54-65,* Springer Berlin/Heidelberg.

Casti, J. 1997. *Would-Be Worlds*, 57-60, John Wiley & Sons, Inc.

Chodos, D.; Stroulia, E.; Naeimi, P. 2009. “An integrated framework for simulation-based training on video and in a virtual world”. *Journal of Virtual Worlds Research, North America.*

Fujimoto, Richard M. 2000. *Parallel and Distributed Simulation Systems*, John Wiley & Sons, Inc..

Hudson , Ken and Degast-Kennedy, Kathryn. 2009. “Canadian border simulation at Loyalist College”, *Journal Of Virtual Worlds Research*, Vol. 2 No. 1.

Koenig, D.; Glover, A.; King, P.; Laforge, G.; Skeet, J. 2007. *Groovy In Action*, 229-276, Manning Publications.

Lang, A andKobilnyk, D. 2009. “Visualizing Atomic Orbitals Using Second Life”, *Journal Of Virtual Worlds Research*, Vol. 2 No. 1.

van Nederveen, S. 2007. Collaborative Design In Second Life, Second International Conference World of Construction Project Management.

Wetsch , L. R. 2008. “Consumer Behavior in Virtual Worlds”, *Journal Of Virtual Worlds Research,* Vol. 1 No. 2.

## BIOGRAPHIES

**SIMON SEE CHONG-WEE** A/Prof Simon See is currently the High Performance Computing Technology Director for Sun Microsystems Inc, Asia and also an Adjunct Associate Professor in Nanyang Technological University. A/Prof See is also the director for the Sun Asia Pacific Science and Technology Center. His research interest is in the area of High Performance Computing, computational science, Applied Mathematics and simulation methodology. He has published over 30 papers in these areas and has won various awards.

Dr. See graduated from University of Salford (UK) with a Ph.D. in electrical engineering and numerical analysis in 1993. Prior to joining Sun, Dr See worked for SGI, DSO National Lab. of Singapore, IBM and International Simulation Ltd (UK). He is also providing consultancy to a number of national research and supercomputing centers. Dr See is also an adjunct research fellow in the National University of Singapore.

**MELVIN KOH CHEE-KIAN** is working as a solution architect in Sun Microsystems and is an associate member of Asia Pacific Science & Technology Center. He received his Degree in Computer Science from National University of Singapore in 2002. Melvin was actively involved in many research collaboration with top research institutes and universities world-wide. His research interests include Grid, Cloud computing and HPC.

**CHE-WING CHEUNG** was born in 1977 in Breda, The Netherlands. He migrated to Singapore three years ago and has been working for Sun Microsystems since. He studied Business Informatics at the Noordelijke Hogeschool in Leeuwarden. His main research interests is in Grid and High Performance Computing related to virtual worlds and its applications.

**DOUGLAS FINNIGAN** has been working in Singapore for the past fifteen years. As a lecturer at Temasek Polytechnic's School of Information Technology & Informatics, he has been involved in numerous 3D-related projects. His main areas of interest are 3D modeling, animation and visualisation; gaming for education; and virtual world technologies.

**JOON JEW LIOW** attained a post-graduate degree (Masters of Technology) from National University of Singapore in the area of software engineering in 2002. He has extensive experience leading teams in the software design and development efforts in the domains of aircraft simulation and user interface design. Currently he is lecturing in Temasek Polytechnic, contributing actively to the curriculum of the software engineering and object-oriented design courses.

# LOGISTICS AND WAREHOUSING TOOLS

# DYNAMIC ROUTING STRATEGIES
# FOR AUTOMATED CONTAINER TERMINALS

Su Min Jeon
Mark B. Duinkerken
Gabriel Lodewijks

Faculty of Mechanical, Maritime and Material Engineering
Department Marine and Transport Technology
Delft University of Technology
Mekelweg 2, 2628 CD, Delft
The Netherlands
e-mail: s.jeon@tudelft.nl

**KEYWORDS**
AGV, Automated Container Terminal, Discrete Simulation, Dynamic Routing

## ABSTRACT

Automated Guided Vehicles (AGVs) play a central role in automated container terminals. To achieve a high productivity of the terminal it is necessary to develop efficient routing methods. This paper studies dynamic routing methods for AGV systems. Dynamic AGV routing offers great advantages over the traditional static routing methods. It can add flexibility, reduce congestion and eliminates deadlock. In this research, a shortest path planning using time windows (SPPTW) concept is used to find the possible alternative routes for each AGV. The optimal route is selected based on the shortest travel time.

Two alternatives are compared in this paper. In the first method presented, the entire route between origin and destination is determined before the AGV starts driving. On large, complex terminals, thus with a large number of nodes, this approach can lead to unacceptable long calculations. To reduce the computation time of the algorithm, a second method is proposed which adds a decision point in the middle of the entire route. The purpose of this decision point is to present a more dynamic situation, where AGVs are allowed to stop during route execution. The decision point can be configured as a parking location on the terminal, not blocking the other traffic, or as a recharging point in real terminal operation.

## INTRODUCTION

The use of Automated Guided Vehicles (AGVs) is one of the most important automation methods in container terminals; it adds flexibility to the operation system and provides easy access to the other facilities.
In container terminals, an AGV travels between quay and yard side to support Quay Crane (QC) and Stack Crane (SC) operations as shown in Figure 1.

Figure 1 shows a terminal operation system which consist of three sub systems: QC operation, AGV operation and SC operation. The AGV system has the center position in this logistic chain; it connects the QC and SC systems. The productivity of the container terminal thus highly depends on the capacity and reliability of the AGV transport system. The operational control of the AGV system has to cope with the transport requirement of QC and SC. AGV operation includes: vehicles dispatching, routing, traffic management and idle AGV positioning. From these, routing is the most important operation factor that directly influences the efficiency of AGV systems. Without an efficient routing strategy, AGV operation will face difficult operational problems. Thus, this paper focuses on the AGV routing problem in container terminals. It is assumed that a fixed guide path layout, which can be described as a graph with nodes and arcs, is used.

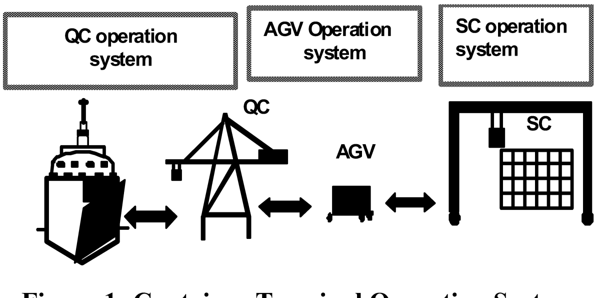


**Figure 1: Container Terminal Operation System**

Routing means: find a route which the AGV can travel from origin to the destination. Routing methods can be divided in two types: static or dynamic. With static routing the route from start location to final destination is determined in advance, during terminal design. The AGV travels through the fixed route. Static routing is easy to control but it is not able to adapt to changes in traffic conditions. In the case of dynamic routing, the routing is determined base on real-time information and it has various possible routes between the same start node and destination node. In this case, the scheduling algorithm has to avoid deadlock and conflict among the AGVs. This study addresses dynamic routing of AGVs, assuming a time window concept which is used for checking the possibilities for the next movements.

The routing problem has been studied by several researchers. Kim and Tanchoco (1991) used the concept of time window graph, which is a directed graph of the free time windows for finding the shortest time route on bi-directional guide path networks. Rajotia et al. (1998) proposed a semi-dynamic time window routing strategy, the principle is quite similar to the path planning method of Kim and Tanchoco (1991). The traffic flow direction is placed on bi-directional arcs, which can only be traveled in one direction at a time. Oboth et al. (1998) suggested a sequential path generation heuristic (SPG) to solve the routing problem. Duinkerken et al. (2006) studied the potential performance of different routing strategies using a simulation model. Möhring et al. (2008) suggested a dynamic routing algorithm using time window concept.

In this paper, the dynamic routing algorithm is based on Möhring's research; it suggests a method to improve the computation time. The remaining section of the paper is organized as follows. In section 2, the suggested dynamic routing method on a mesh type layout for AGVs is presented. Section 3 describes the simulation model used in the experiments and results. Concluding remarks are drawn in section 4.

## DYNAMIC ROUTING METHOD WITH TIME WINDOW FOR AGVS

This section introduces how to plan the routes for vehicles dynamically. The objective of the dynamic routing is to maximize the productivity of the system by minimizing the travel times of the AGVs.

### Fixed guide path layout

Before we move to the dynamic routing issues, we shortly comment on the guide path assumptions of this study. The vehicles travel along fixed guide paths to perform the transport task. The guide path layout can be represented as network consisting of nodes and arcs. Figure 2 shows the guide paths of this study which called a mesh layout (Duinkerken, 2006). The mesh layout can be divided into 3 areas: berthing, parking and stacking. The nodes in the berthing area are used for QC operation, stacking area nodes are the locations for SC operation and parking area nodes are the decision nodes for the routing method. A mesh-type layout is used in practice in the Hamburg container terminal; an AGV can use shortcuts to travel on the shortest possible route from its origin to the final destination. The shortest possible route is calculated as the sum of absolute difference in x- and y-locations.

In the mesh layout, a node defines the start and endpoints of arcs. Every arc has a set of reserved time intervals. Reserved time intervals have to be mutually disjoint, because overlapping reserve times means that the corresponding AGVs can collide on that arc.

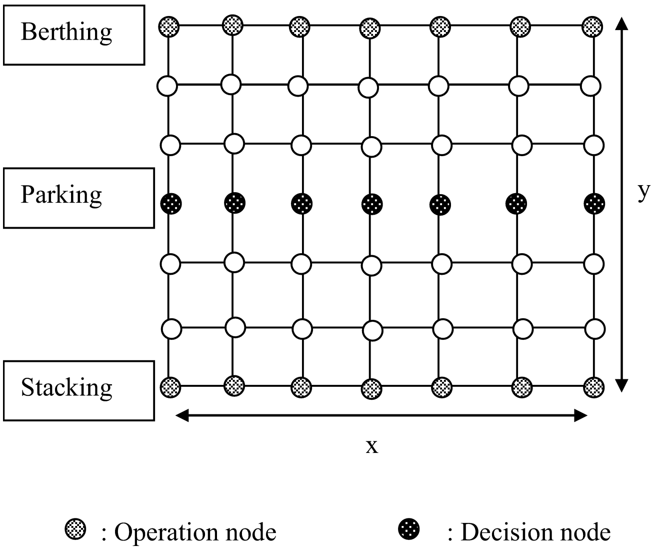


**Figure 2: Example of an Mesh Layout**

### Dynamic routing procedure

The methodology of this paper is based on Möhring algorithm to find the shortest time path. Möhrings et al.(2008) defined an algorithm that determines a new route for an AGVs while taking into account the free time intervals on the arcs that defines the route. Because each new route has to respect the reserved time windows (from previous routes) on the arcs, traffic problems like deadlock can not occur. The goal of Möhrings algorithm is to compute the shortest path with respect to cost (cost = transit time + possible waiting time) that respects the given time windows. The main advantage of this algorithm is that the time dependent behavior of the AGVs is fully modeled, such that both conflicts and deadlock situations can be prevented already at the time of route computation. The computation time for this algorithm is related to the number of arcs in the layout and can be long in the case of a terminal with realistic dimensions.

The routing method in this paper is a adapted version of Möhrings algorithm. First, instead of allowing the vehicle to wait on every arc (as long as it does not conflict with the reserved time windows), in our interpretation it is assumed that each route can be travelled without stops. If waiting during a route occurs it will be only at the starting node of a route. If the routing algorithm can not find a non-stop route, a small time-step $t_i$ is added to the possible leaving time at the startnode and a new route is calculated; this process is repeated until an acceptable route is determined.

The second modification is the addition of a decision point in a route. Generally, the starting node and the destination node of an AGV route are determined by the location of each QC and SC position. When an AGV completes its current task, it will receive a new task, which includes starting and destination position. However, for larger terminal sizes and thus a large number of arcs, the algorithm computation time can become unacceptable high. Therefore, we added decision nodes in the middle area of the mesh layout to reduce the complexity of computation for each route. Instead of calculating a single route, two routes are calculated, from start to decision node and from decision node to destination. The sum of the computation times for the two shorter routes is less then the computation time for the complete route. Beside that, the extra possibility of waiting at the decision point might give an advantage which is discussed later.

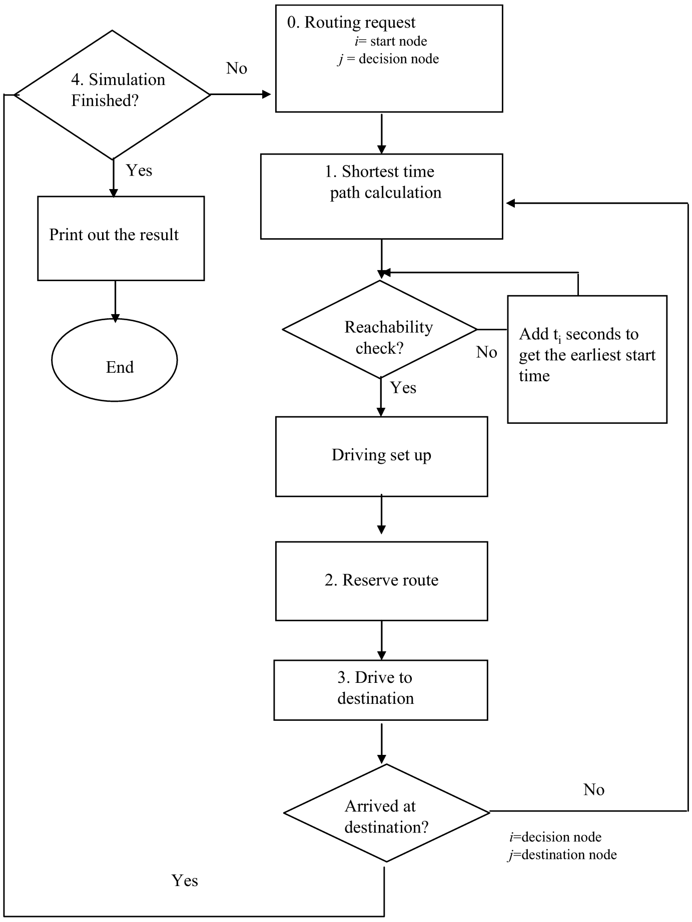


**Figure 3: Procedure of Dynamic Routing**

Figure3 shows the dynamic routing procedure of this study. The five steps of the procedure are:

Step 0. During initialization and after task completion, the AGV receives a destination node.

Step 1. Calculate the shortest time path respecting the time windows. If the shortest time path can be travelled non-stop, go to step 2, otherwise wait a predetermined time $t_i$ and retry step 1.

Step 2. Calculate the driving time for each arc on the route and reserve the time windows.

Step 3. Drive the route. If the last node is a decision node then move to step 1 otherwise move to step 4.

Step 4. If the simulation state satisfies the termination condition, print out the result, otherwise move to step 0.

## SIMULATION STUDY

### Simulation scenario

The goal of the simulation study is testing the performance of the routing methods for AGVs. A simulation program was developed using Delphi 7 and the TOMAS simulation package (Veeke and Ottjes, 2002) which allows for object oriented discrete event simulation. The container terminal was modeled with one berth of the length of 300m, 6 QCs and 13 YCs. The fixed guide path layout of the simulation model is similar to the mesh layout in Figure 2.

We assume that nodes in the parking area are decision nodes and all vehicles can pass only one decision node. At the decision node, AGVs can wait and determine the route for the remaining part to the final destination. The decision point for each route is selected as the parking node closest to the straight line connection between origin and destination. Another assumption of this study is that no time is taken for the QC and SC operation which means that we focus on the driving and waiting times of the AGVs. For each AGV, a destination and start point are drawn from a uniform distribution. AGVs travel with constant speed. The AGV router performs the task of implementing the suggested algorithm on the layout: as soon as an AGV with a new job announces itself, the AGV router determines the shortest time path with respect to costs taking into account blockades by other AGVs, blocks the route and announces the route to the AGV. When an AGV reach a QC or SC, it draws a new destination crane. While the AGV travels the route, performance indicators are measured.

### Simulation results and discussion

In this section, we present the result of the simulation experiments. The simulation program runs on a computer with a 2 GHz Pentium IV processor, 1 GB RAM and Windows XP. Because the difference between longest and shortest route is large, the effects of the initial positioning of the AGVs disappear fast, therefore a warm-up period is not needed. To allow each connection to be travelled multiple times, the chosen simulation runtime was 3600 seconds, resulting in about 12 jobs per possible connection on the average. The iteration time $t_i$ was set at 10 seconds; a too short value would result in too many iterations, thus a longer computation time, while a too long iteration time time would result in loss of transport capacity.

The simulation experiment was repeated 10 times for each scenario, using different randomseed values. The number of AGVs went from 10 to 50 with increments of 10. The simulation model implemented the two routing methods, to compare their performance:

- 1 step: a route plan from start to destination is determined before the AGV starts driving.
- 2 step: a route divided into 2 parts: from start to decision node and from decision node to final destination.

Figure 4 shows the average job performance of AGVs. The performance of the 1 step method is higher than the performance of the 2 step method. A simple t-test proved that the difference in performance is significant.

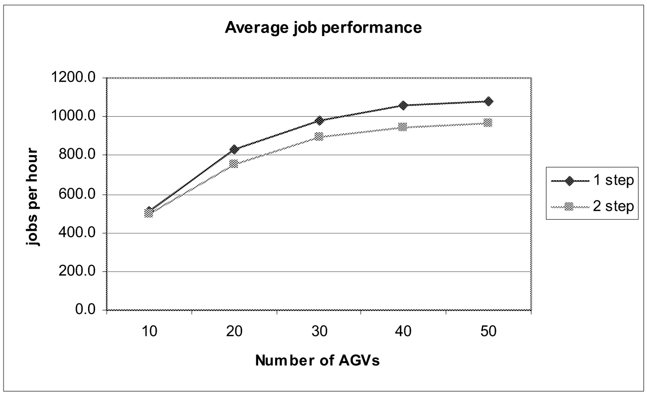


**Figure 4: Average Job Performance**

However, the average computation time of the 2 step method is lower than in the 1 step algorithm, as shown in Figure 5 and also proven to be significant. It shows that the decision point addition is helping to reduce the algorithm computation time which also impact to real operation of AGVs. This is especially the case for lower number of AGVs. Figure 5 suggests that with 50 AGVs the advantage of the 2 step method disappears.

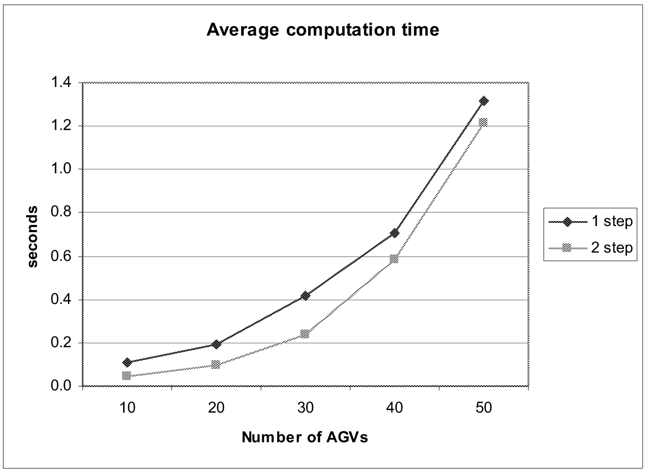


**Figure 5: Average Computation Time per Route**

Another performance indicator is the average waiting time per job. There are two types of waiting times collected during the simulations: the waiting time at the starting points and the waiting time at the decision nodes. Figure 6 shows the average total waiting time which is the sum of the two types of waiting, for different number of AGVs. For each number of AGVs, the left bar shows the waiting time for the 1 step method and the right bar shows the sum of waiting at starting node and decision node for the 2 step method. It was found that addition of the decision node does not largely influence the total waiting time of the AGV.

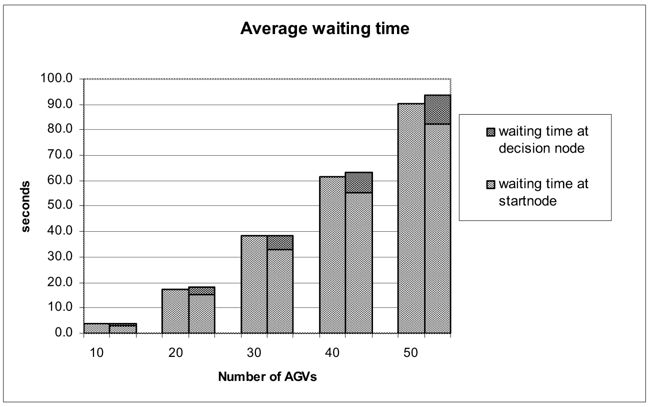


**Figure 6: Average Waiting Time**

From these results, although the job performance of the 2 step method is lower than the 1 step method, it can be concluded that the 2 step method is the more practical routing strategy considering the computation time of algorithm and the average waiting time.

## CONCLUSION

This paper studies dynamic routing algorithms for AGVs in container terminals. AGV routing is one of the most important strategies to get a high productivity in container terminals. The goal of this study is to find the shortest traveling time route for each delivery demand. We implemented an algorithm based on shortest path planning using time-windows. To cover the weak point in this algorithm, we added decision nodes in the parking area. The purpose of this decision nodes addition is to allow stopping during traveling and to reduce the computation time.

Through a simulation study, the performance of the algorithm with decision points was compared to the original algorithm. It was shown that the decision node has potential to improve dynamic routing. However, much more candidate nodes are needed for selecting a decision node.

In this study, we predetermined the decision node for each route, which might not be optimal. This could be the reason why AGVs at the decision node take a long waiting time. For further study, we suggest two types of alternatives to select a better decision node:

1. Grouping the potential decision nodes considering the location of QCs and YCs and selecting the decision node which returns the best result.
2. Considering the active jobs for each origin and destination, selecting the decision node with the lowest workload.

The suggested algorithms must be tested in a simulation environment which is much more similar to the real ship operations before the algorithm can be applied to practice.

## ACKNOWLEDGMENT

This work was supported in part by the Next Generation Infrastructures Foundation.

## REFERENCES

Duinkerken, M. B., J. A. Ottjes and G. Lodewijks, 2006. "Comparison of routing strategies for AGV systems using simulation." *Proceedings of the winter simulation conference*, pp.1523-1530.

Kim, C. W. and Tanchoco, J. M. A., 1991. "Conflict free shortest time bi-directional AGV routing." *International Journal of Production Research*, 29(12),pp.2377-2391.

Möhring, R. H., E. Köhler, E. Gawrilow and B. Stenzel, 2008. "Dynamic routing of automated guided vehicle in real-time." *Mathematics - key technology for the future*, Springer Berlin Heidelberg.

Oboth, C., Batta, R., Karwan, M., 1999. "Dynamic conflict free routing of automated guided vehicles." *International Journal of Production Research*, 37(9), pp.2003-2030.

Rajotia, S., Shanker, K., Batra, J. L., 1998. "A semi-dynamic time window constrained routing strategy in an AGV system." *International Journal of Production Research*, 36(1), pp.35-50.

Veeke, H. P. M. and J. A. Ottjes., 2002. "TOMAS: Tool for object-oriented modelling and simulation." *Proceedings of the business and industry simulation symposium*, Washington D.C.

# COMBINING SCRIPTING AND COMMERCIAL SIMULATION SOFTWARE TO SIMULATE IN-PLANT LOGISTICS

Tim Govaert, Sven Neirynck, Sofie Van Volsem and Hendrik Van Landeghem
Department of Industrial Management
Ghent University
Technologiepark 903
BE-9052 Zwijnaarde, Belgium
e-mail: {Tim.Govaert,Sven.Neirynck,Sofie.VanVolsem,Hendrik.VanLandeghem}@UGent.be

**KEYWORDS**
simulation, automatic model generation, Python, Plant Simulation

## ABSTRACT

In this paper we describe the use of a commercial discrete event simulation package (Siemens 2008) combined with a custom program, written in the programming language Python (Martelli 2006). Combining these two makes it possible to automatically generate a model for assembly line logistics simulation. The different stations of the assembly line, their connections and the storage near the assembly line were generated within seconds. A huge amount of time was saved in comparison to manual generation.

## INTRODUCTION

In the truck assembly industry, in-plant transportation should be handled in the most cost-efficient way. Taking into account the fact that forklifts fail when it comes to efficiency, the truck industry started investigating the use of automatic transportation systems such as overhead conveyors or Automated Guided Vehicles.

Implementing an automatic system on factory scale requires extensive research. Cottyn et al. (2008) executed a feasibility study to investigate the possible gains and necessary investments. They suggested to build a simulation model to discover possible pitfalls of the system and to be able to dimension more in detail the amount of drop-off stations, pick-up points and carriers.

In the digital factory concept, the product and process planning can be designed and improved on all levels, by using various simulation processes. A broad overview of applicable areas for simulation is given by Kühn (2006).

In order to create a simulation model that is easily adaptable and flexible in use for the particular problem of simulating the in-plant logistics processes for truck assembly, two problems were encountered:

**Modeling the assembly line** A factory can contain a huge amount of workstations, which can change very often. Therefore, it was decided that the generation of the workstations in the model should be automatic and very flexible. In this way, different factories or subparts of one factory can be easily generated and simulated. A big increase in model flexibility can thus be obtained.

**Modeling storage buffers at the border of line** The huge diversification of the customers needs results in a huge variety of parts. These parts need to be stored at the line, in order to be consumed when the corresponding chassis passes the workstation. The parts can be bulk fed or brought at line in kits (Limère and Van Landeghem 2009). In each situation, the amount and configuration of buffers will be different. An automatic generation of these buffers could drastically decrease the modeling time.

In the next paragraph, we present the solution method. Thereafter, preliminary results of a practical case study in the truck industry, using the proposed method, are given.

## METHOD

### Simulation environment

Two options exist for implementing simulations:

- develop a dedicated computer program that implements a specific simulation problem
- use a commercial simulation package to model the simulation problem at hand

In an effort to try and combine the advantages of both approaches, we propose to use a commercial simulation package[1] to simulate the plant (Siemens 2008), while generating large parts of the model via a custom program. This program is written in the programming language Python (Martelli 2006); ASCII files are used to interface between the two distinct environments.

[1] Tecnomatix Plant Simulation

Python is an interpreted, high-level programming language. Like all scripting languages, Python code resembles pseudo code. Python's syntax is clear and readable. The way Python's syntax is organized imposes some order to programmers. Experts and beginners can easily understand the code.

### Model Components

A production plant as described in the introductory section can be divided into objects belonging to four categories:

- line supplier
- transportation system
- border of line (BOL) storage
- assembly line

*Line supplier*
A line supplier is an entity that delivers parts to the transportation system. These entities can be warehouses, pre-assemblies or supermarkets. A supermarket is a logistical area where kitting takes place.

*Transportation system*
The transportation system takes care of the transport of "parts" between buffers. The origin buffer is always the output buffer of a *line supplier.* The destination buffer is the *border of line.* Identical parts, i.e. parts with the same part number, are put in a container. The transportation system transports this container over transportation tracks.

*Border of line storage*
The border of line (BOL) is modeled as different buffers. The parts stored in these buffers are ordered[2] (FIFO queues). On the transportation track inside the station an "offload" point is present where the container on transport can be moved to a buffer.

*Assembly line*
An assembly line is modeled as a series of connected stations. A station is where the assembly of the trucks takes place. A station receives a partially completed chassis from the preceding station. At the station parts are added on to this chassis. These parts are retrieved from BOL buffers locally to that station. After a fixed amount of time (the takt-time), the chassis is moved to the next station.

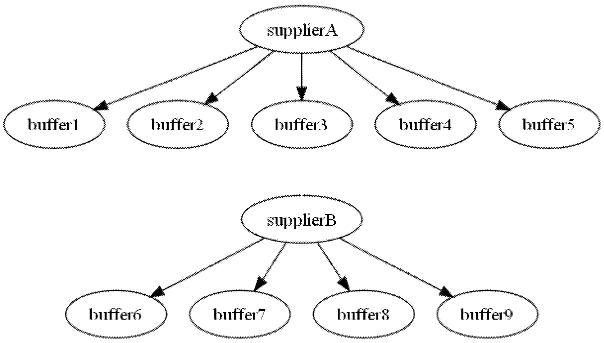


Figure 1: Relation Buffers and Suppliers

### Model generation

Two types of objects are automatically generated: stations and buffers. To automatically generate an assembly line not only the stations need to be created but also the connections.

*Assembly line generation: creating stations*
Added functionality can be programmed in Plant Simulation by using so-called "methods" Siemens (2008). Methods are like functions: they can have input, output, program logic and can perform various tasks.
In Plant Simulation data objects exist such as tables and queues. A table object called *FactoryLayout* is created. This table is a database describing all stations. For each station there is an entry with the following information:

- station name
- x and y coordinate of the station's location center
- type of station, determined by number and place of BOLs
- preceding station
- a boolean value indicating whether transportation tracks have to be generated

A method was written which uses the *FactoryLayout* table as input. For each entry in the *FactoryLayout* table, a station at the given coordinates is created. If necessary, transportation tracks are also created.

*Assembly line generation: creating connections*
Two types of connections need to be created:

- Connections between output of one station to the input of the next station to model the flow of the chassis through the factory.
- Connections between the tracks of the transport system:
  - The tracks need to form a closed loop

[2] Parts are ordered in a container, and containers are ordered in the buffer.

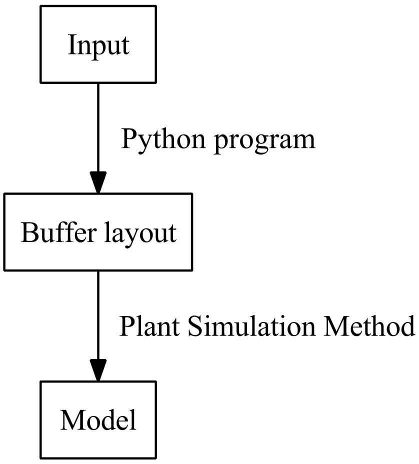


Figure 2: Data flow

   - Junctions to reach the drop-off points inside the stations need to exist

Not all connection information can be stored in the *FactoryLayout* table. This table only stores the stations predecessor. Examples of extra connections are: feeder lines, merging of two lines, connectivity of the last station to the exit of the factory. The remaining connections are put in another table called *Connections*. Another "method" uses this table to create the remaining connections.

*Buffer generation*
The buffers from the border of line as well as the assignment of these buffers to the different parts is dynamic. It varies between different simulation runs.
One of the goals of the eventual model is to explore different "kitting" combinations. Each combination has its impact on the buffers. A suitable model therefore needs to be dynamic in buffer allocation. To assign buffers to the stations we need to have the information about what trucks need which parts at which station. We need to process this information and calculate the buffer assignment. This buffer information is then fed into the model.
Information about the trucks is needed. A Python program transforms this into output which contains information about the buffer layout. A Plant Simulation "method" will use this buffer layout information to create the model. (See Figure 2)
During the simulation, parts will be consumed at the stations. As a consequence buffers need to be refilled. Some parts will come from a pre-assembly, others will be a kit from a supermarket and some can just be retrieved from a warehouse. Thus, information is needed on the in-plant origin of the different parts.

**Input:** The Python program requires the following input:

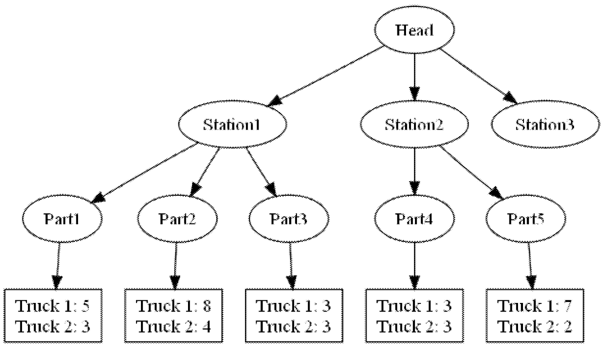


Figure 3: Example tree

- *Parts Requirements List:* A file describing for each truck all the parts needed for assembly and the station where the assembly takes place
- A file describing the in-plant origin for each part

**Output:** The external[3] program written in Python transforms this information into the layout of the buffers. It generates the following output:

- for each station: the number of buffers;
- for each buffer: a partnumber, supplier and the number of parts to be assembled for each truck.

**Algorithm:** The algorithm to calculate the buffer layout consists of two phases. In phase 1 the input is parsed into a tree. In phase 2 the tree is written out into individual files which will serve as input for Plant Simulation.

**Algorithm 1** The two phases

```
Create empty tree
for Each Line in Parts Requirements List do
  Parse station name
  Add truck, station, part to the tree
end for
for Each station in the tree do
  Count parts
  Write number of parts to file
  for Each part in the station do
    Create buffer output file
    for Each truck do
      Write truck and number to the output file
    end for
  end for
end for
```

We use the following I/O files:

**"Stationname"."L/R".txt** contains the number of buffers at this station. (Some stations have a left

[3] External to Plant Simulation

and right side indicated by an extra character L or R).

**"Stationname"."L/R""buffernumber".txt** is the table stating the quantity used for each different truck.

**"Stationname""buffernr".supplier.txt** is a file with the in-plant supplier's address and the partnumber for that buffer.

Inside the station there is a "method", *configure*, which configures the station. This method looks for the files *"Stationname"."L/R".txt* and creates the necessary number of buffers. For each buffer it reads the file *"Stationname"."L/R""buffernumber".txt*. It also reads in the *partnumber* and *supplier*. This information is needed by the model during simulation.

## CASE STUDY

We implemented these two techniques to create models for the supply chain logistics for a large European truck factory.

### Assembly line layout

The *FactoryLayout* table consists of over 100 entries. Each entry represents a station of the assembly line. Three different Plant Simulation station objects are used. The *connections* table consists of 64 extra connections. Some tracks and warehouses are manually placed in the model. Generation of the model takes less than a minute on a standard desktop.

### Buffer layout

Using the information of 1264 trucks assembled at the plant, the Python program created 7865 files in a few minutes. The Python program created a file for each station and two files for each buffer in the stations.

### Future usage of the model

After creating and validating the model, simulations will be run in order to study lead times, queues, waiting times, possible problems and the influence of unexpected circumstances such as breakdowns, shortage of materials, etc.

## CONCLUSIONS

We have successfully created an interface between a commercial simulation package (Plant Simulation) and a scripting language (Python).
Within Plant Simulation we are able to generate the factory layout in a very flexible and cost effective way.
Using the scripting language Python we are able to dynamically create a model (buffer configuration) using a few input files. This would have not been possible using a conventional approach without resorting to an external programming language.

## REFERENCES

Cottyn J.; Govaert T.; and Van Landeghem H., 2008. *Alternative line delivery strategies support a forklift free transition in a high product variety environment.* In *Proceedings of the International Workshop on Harbor Maritime and Multimodal Logistics Modeling and Simulation.* Campora S. Giovanni, Italy.

Kühn W., 2006. *Digital factory - simulation enhancing the product and production engineering process.* In *Proceedings of the 2006 Winter Simulation Conference.* 1899–1906.

Limère V. and Van Landeghem H., 2009. *Cost model for parts supply in automotive industry.* In *Proceedings of the 16th European Concurrent Engineering Conference.* Eurosis, Bruges, Belgium, 120–125.

Martelli A., 2006. *Python in a Nuthsell.* O'Reilly, 2nd ed.

Siemens, 2008. *Tecnomatix Plant Simulation 8.2 Step-by-Step Help.* Siemens.

## AUTHOR BIOGRAPHY

**TIM GOVAERT** is academic assistant at the department of Industrial Management at Ghent University. He is currently working on his PhD in in-plant logistics. His research interests lie in material handling, facility logistics and lean manufacturing. Tim already completed several projects within the automotive and truck industry.

**SVEN NEIRYNCK** graduated as MSc in Computer Sciences at Ghent University in 1997. After 10 years of working as a systems engineering manager, he currently divides his time between IT consultancy with expertise in storage, archiving, data security and HPC systems; and research in the area of simulation at Ghent University.

**SOFIE VAN VOLSEM** received a MSc degree in Chemical Engineering from Ghent University in 1998 and a PhD in Engineering Sciences from the same institution in 2006. She worked in industry as a process & quality engineer before returning to academia. After being with the University of Antwerp for 6 years and the University College of West-Flanders for nearly 2 years, she currently holds a post-doc position at Ghent University. She teaches Quality & Industrial Statistics. Her research interests are quality management, quality and reliability issues in supply chains, management applications of metaheuristics.

**HENDRIK VAN LANDEGHEM** is Professor at the department of Industrial Management at Ghent University. He is an expert in the area of logistics and their application in business processes. He advises companies in their choice and implementation of their logistic organization and production control systems. He is Fellow of the European Academy of Industrial Management (AIM) and member of the Institute of Industrial Engineering (IIE) and of the European Operations Management Association (EurOMA). He is since 2007 Fellow of the World Confederation of Productivity Science.

# CONDUCTING THE SIMULATION OF WAREHOUSING SYSTEMS

Christian-Andreas Schumann
Noemi Nikoghosyan
Andreas Rutsch
University of Applied Sciences Zwickau
Institute of Management and Information
08056 Zwickau, Germany
E-mail: Christian.Schumann@fh-zwickau.de

**KEYWORDS**
Continuous simulation, optimization, scheduling, software cost analysis, system engineering

## ABSTRACT

The optimization of inventory management and the whole ordering process realization is a costly and time-consuming process. It consumes almost 60% of all warehouse activities. Various methods of products positioning and process chain shortcuts lead to essential improvements. In this paper some of those methods are described. In order to determine potential optimization several simulation tools are examined and evaluated. Upon the initially indicated requirements only two simulation tools remained. A ‘small’ size model with few objects was created for those two remaining tools in order to get a deeper conception. Afterwards the customer was provided with the assumption of the remaining tools’ general features comparison.

## INTRODUCTION

Companies doing business follow their goals and principles in order to develop and achieve competitive advantage in the conducted field of business. For this reason those companies are dealing with innovative technologies, in order to get deeper in the logistics systems, which support the overall processes. Here are the points that companies are dealing and mostly interested to achieve:

- Development of advanced storage systems to strengthen position in the business area
- Definition and design new warehousing systems or distribution centers
- Grant a certain level of flexibility
- Increase the demand orientation, etc.

Mathematical description of a logistics system is aimed for analytical solution and its complexity as well as realism is quite limited. Computer models, on the other hand, can face the complex formulations of models and can prove the analytical studies. For this reason simulation can be used, which apart from supplying full description of the logistics systems, provides a specific procedure which formulates and tests the hypotheses. Thus, gained results can be realized, that are implemented in real world. By using a simulation tool, companies can be supported with the following logistics concerns:

- New Warehouse Design
- Definition of Standard Parameters for warehouse design based on business requirements
- Goods Flow Simulation and Optimization in actual and futures warehouses
- Existing Warehouse Layout Optimizations
- Documentation
- Supporting further Business Development

The main concept is to study those logistics concerns, describe them and obtain optimal solutions. It is preferable to start all the operations with an already existing logistics system, so that the results can be compared with reality, already working, system. Therefore the starting point is to develop and optimize operations based on the status quo. That is why, at the beginning of the optimization the following main aspects are going to be considered

(1) inventory management, which is about loads positioning and
(2) processes, which realize the flow in the system.

## INVENTORY POSITIONING AND MANAGEMENT

A static layout of a warehousing system is defined and examined. Several aspects on how to place the loads in storage can be studied, taking into consideration the fact of different policies related to placement. The research pays special attention to a customer order process with the purpose of improving warehouse operational efficiency. Optimization can be achieved by implementing several changes on the load positioning in a warehousing system, such as:

- Loads placement using Pareto principle, also known as the 80-20 rule, which is one of the common positioning variants. The positioning can vary depending on whether entrance and exit docks are located at the same place or not. In the picture below there are shown some strategies of loads positioning. In warehousing systems loads are usually categorized in groups, where categorization factor is loads popularity, upon order frequency. Therefore a warehouse can be divided into positioning zones A, B or C (Figure 1), which represent the number of vehicle visits. So, positions in zone A have highest visits, and positions in B zone are visited more often than in zone C. (Merkuryev et al. 2009)

- Positioning the same kind of loads in various places, that is locations in different aisles, in a warehousing system. In this case vehicles will have a chance to choose a more appropriate path, which is usually considered as the shortest one, in a system to travel. In that case while realizing an order vehicles do not need to travel through the whole warehouse to get the needed loads.
- While in most cases there are several loads that are supposed to be picked in one run of a vehicle it is good to store some specified loads near each other, where specified is in the understanding of logically interconnection. This logical interconnection specification can be assumed from lifelike conceptions, which is usually the fact, that there are goods that are usually being ordered together by a customer, like chairs and tables, pens and pencils, etc. These specifications can be also assumed through simulations.

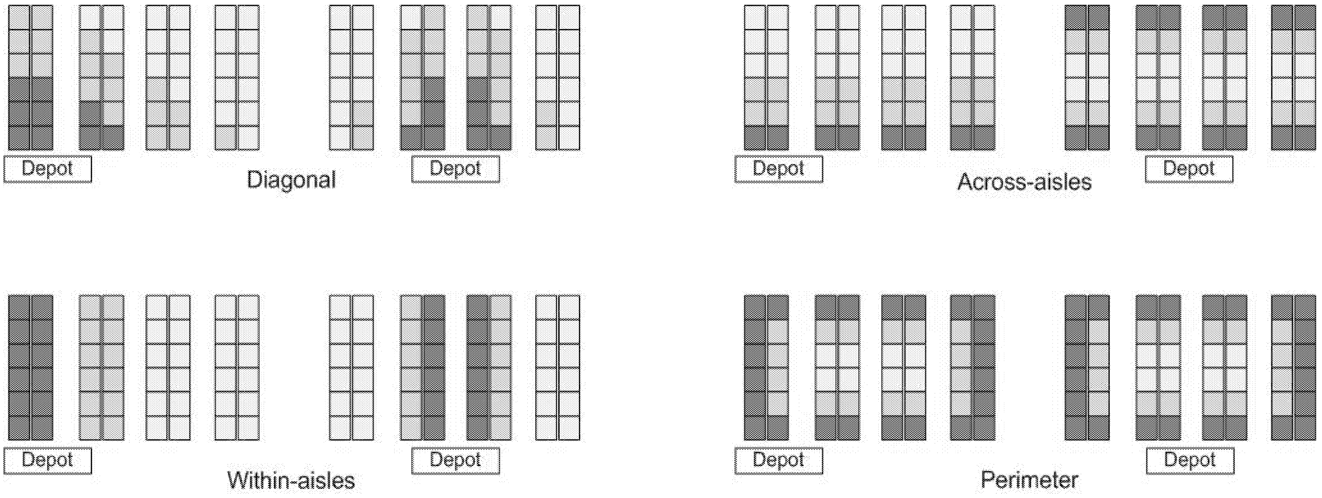


Figure 1: Storage methods, where dark blue A positions, medium blue B positions, white C positions (Merkuryev et al. 2009)

The target of the case study is to define combinations of various strategies, by which the minimization of travel distance can be achieved, thus accelerating the process within the warehouse, particularly order picking process. The latter is the most laborious of all warehouse processes and it may consume almost 60% and can be improved by maximum 68% of all labor activities in the warehouse (Merkuryeva et al. 2006). Theoretical views come out by the above mentioned factors and many others may optimize the system and can be pre-checked by simulation.

## PROCESS SHORTCUTS AND OPTIMIZATION

When the inventory is initially managed, another important option, which is supposed to be studied, is processes within a warehousing system. The processes organize the flow in a system. For any action in a warehouse a process should be described. In general, there are the following processes:

- Receiving
- Putaway
- Storage
- Order picking
- Shipping

Receiving is the setup for all other warehousing activities. Products need to be received properly, otherwise next processes will face difficulties. But on the other hand order picking operation is identified as the highest priority activity in the warehouse for productivity improvements. A recent studies show that order picking takes a large part of the warehouse operating costs and errors, due to the labor intensity. The introduction of new activities, such as just-in-time, quick response, new marketing strategies etc. as well as quality improvement and customer service make order picking process more complicated. Therefore storage and accuracy requirements have increased. Order picking is a process which seeks for products in specified locations according to a customer order. Thus, corresponding changes of inventory will affect and may optimize the processes of warehousing system.

But what is more interesting to consider here, is the fact, that in some peculiar cases, the simulation model runs not all the processes of the mentioned process chain. In some defined cases the model can drop or skip an operation or a general process using so-called shortcuts. Thus the following main shortcuts can be defined:

- Traditional Receiving
- direct secondary putaway
- direct primary putaway
- Cross-docking (Figure 2)

All shortcuts are worthwhile to be examined, because of higher efficiency, but the obvious effect is achieved certainly with cross-docking. When an order is being realized before sending the vehicles through the chain of processes, for corresponding actions, the system forces it to check, whether at that exact time the same kind of loads, which stay in the order list, are at receiving stage in the system (note that in some warehousing systems shipping and receiving docks have different allocations). In that case those loads would be just taken from a dock, where they are arriving, to the shipping dock, where the appropriate order is to be realized. Thus cross-docking will take place.

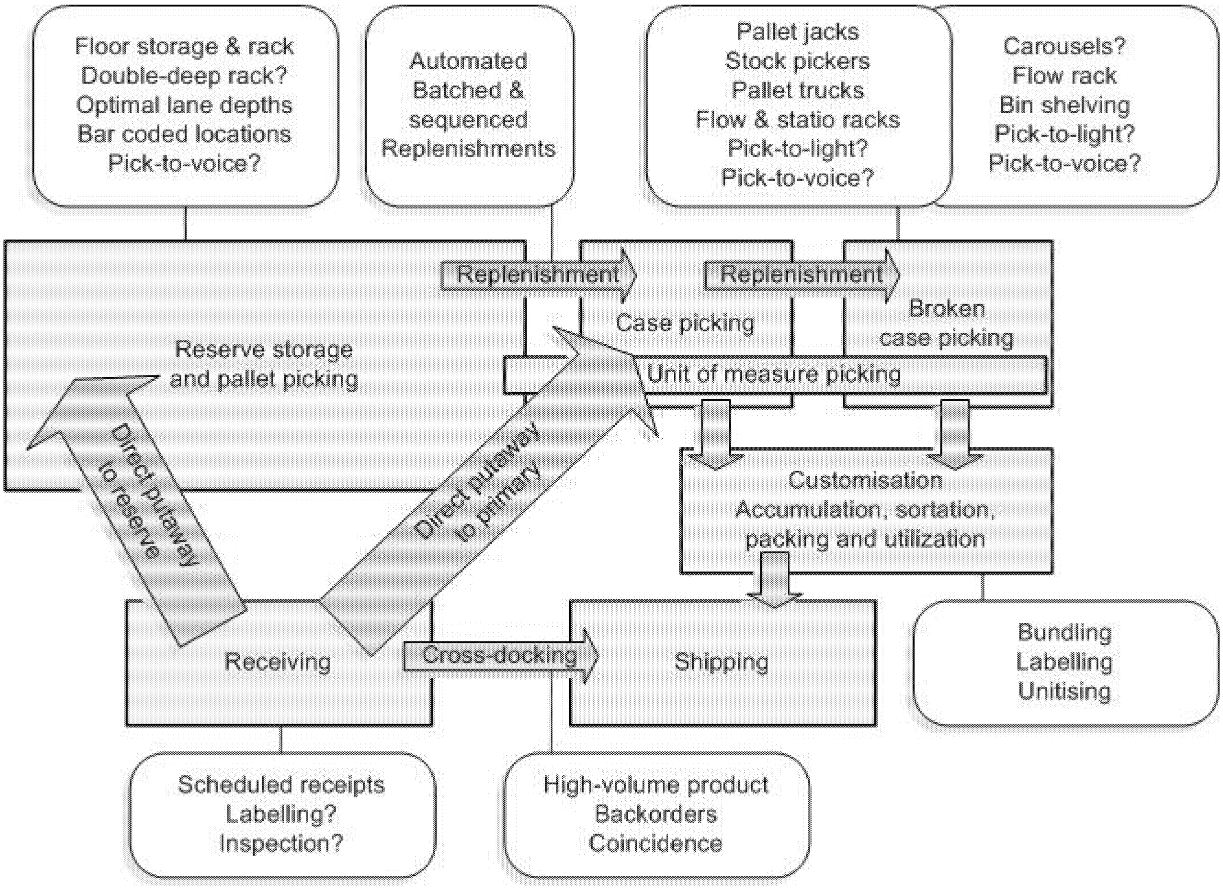


Figure 2: Traditional U-shape [distribution centre] configuration (Frazelle 2002)

## SIMULATION TOOLS EXAMINATION

As soon as the static layout of warehousing system is described, all the information about the objects in the warehouse is defined, hence the simulation tool can be chosen. Several simulation tools were evaluated in order to get the 'proper' tool for the given, described, warehousing system environment. It is the warehousing system's model description that provides information, what features the simulation tool is supposed to capture, so that the created model fits requirements. For that reason following main factors have been weighted for evaluations:

- Functionality (main issue to achieve the scope of the project)
- Cost (cost-benefit ratio is important and therefore assessed high)
- Software support (maintenance and support are important facts for long-term strategies)
- Spread and reputation

By initial market research the following tools were chosen to be evaluated: Arena, AutoMod, Class, Enterprise Dynamics Logistics Suite, Flexsim, Promodel, Showflow, Simcad, Simul8, and Tecnomatix. Besides of short tests and trial versions of the tools, also other sources were used for the evaluation, such as vendor's websites and articles from diverse sources. Finally the following scoring assumptions were made for the above mentioned main factors:

- In terms of functionality, AutoMod was the clear winner in the evaluation. Main strengths/ contributing factors were the ability to design virtual warehouses and to model their operations.
- In terms of initial costs, Showflow was the clear winner of this evaluation.
- In terms of support, at the first view the overall range of tools showed a good performance, while only Showflow seems to fail along the line.
- For the spread and recommendation information from professional journals, which rated the simulation tools via tests, was used. Information concerning to the spread was not always indicative, while academic licenses are not explicit excluded in the fact and figures.

## AUTOMOD VS. ENTERPRISE DYNAMICS LOGISTICS SUITE

After the price proposal, which was to estimate the costs, over the next five years, two simulation tools were chosen for further studies – AutoMod and Enterprise Dynamics Logistics Suite (ED Log). In order to achieve a deep conception of the chosen tools an initial ‘small’ model for each has been created (Figures 3 and 4). Certainly each tool has its own advantages and disadvantages, especially depending on the described model, that is in some cases the features of one tool are more appropriate than the one of the other (Yuriy and Vayenas 2008). The initially described model faces some corresponding changes for each tool and adapts to them. The general features of AutoMod and ED Log tools under comparison are:

- Working environment
- Main objects to create a model
- Simulation

### Working environment

The model creation is detailed and difficult procedure and a proper working environment may support in this greatly. AutoMod uses several applications to define or create a model and runs the simulation whereas ED Log contains one general working environment. Menu in ED Log is organized in an apprehensible and easy access way, which offers as well wide range of shortcuts. The objects are located in one panel – library tree, and creation process takes place by dragging them into the model. AutoMod, on the other hand, creates each object through defining the parameters and then only places them in the model. The latter tool uses several working panel which are located in several systems. Thereby to proceed from one object to another, during model creation, working panel, that is system, has to be changed.

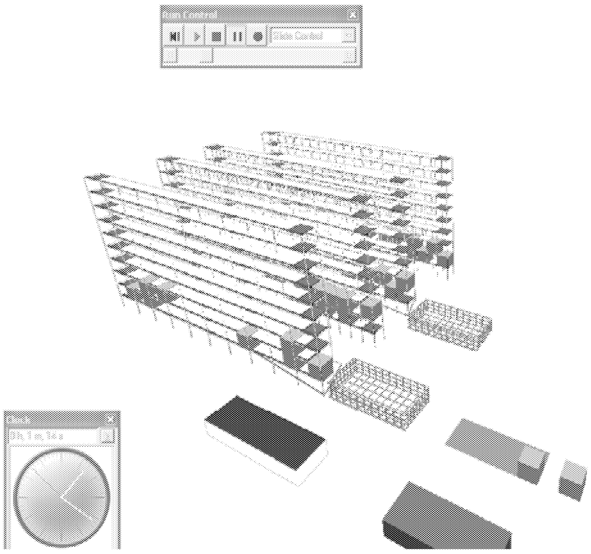


Figure 3: A simple model created by Enterprise Dynamics Logistics Suite

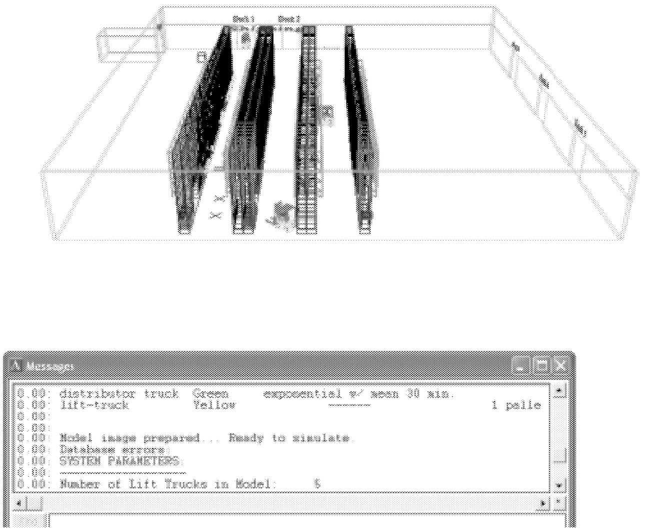
It can be concluded that in the understanding of the working environment ED Log provides an easier access to the tools and objects in the model, thus making definition of a model simple.

### Main object to create a model

As main objects, including their functionalities, almost in any model are:

- Products/loads
- Resources
- Vehicles

Connections of those objects are represented through a logic, which is described using scripts. One of the important aspects is about creating the objects in the model. Here tools behave differently. In ED Log it is simpler to define an object, only one at a time. Though AutoMod requires more time to complete object’s description, through definition of the object it allows to mention the quantity of the created object to appear in the model. In most warehousing systems this detailed description is not necessary. The existence of that information cannot really affect the results, by making several corresponding assumptions the ‘extra’ data can be removed. But on the other hand, there are case studies, that contain attributes making the model true-to-life and easy to program.



```
0.00: distributor truck  Green     exponential w/ mean 30 min.
0.00: lift-truck         Yellow          -------              1 palle
0.00:
0.00:
0.00: Model image prepared... Ready to simulate.
0.00: Database errors:
0.00: SYSTEM PARAMETERS:
0.00: ------------------------
0.00: Number of Lift Trucks in Model:     5
```

Figure 4: A simple model created by AutoMod

AutoMod provides sizeable set of attributes for the products definition, which supports complete description of the product. ED Log assists with wide choice library for products visualization. In contrast AutoMod owns an application for objects’ design and various figures can be created and stored for further use.

Beside of the fact, that ED Log has a better environment for scripting AutoMod provides flexibility to the model specification. Both tools possess their own scripting language. In ED Log each object is described by itself and has its own script which is placed in the object. Processes and connections between those objects are given by channels. In AutoMod all the processes and interconnections between objects are written in a general script by using different functions from the library or by creating new ones, to achieve flexibility.

In case of creating operators or resources in the model, which are supposed to represent a machine or a man, who does some defined work, for example inspects the products, adds a detail or does packing, etc. in AutoMod, rather ED Log, is more likely to be a machine, though working times can be given, but the operator cannot wander in the created system as in ED Log.

Objects of the category trucks or vehicles contain almost the same functionality in both tools, such as capacity, velocity, size, load/unload time, acceleration/deceleration time, etc. However, AutoMod is complex in creating vehicles, since the path through which they travel is supposed to be defined. The logic of movement from any point to another has to be described in a working list. In case a vehicle has no work to do, then it is looking in a defined parking list for a place to 'rest'. Though it should be mentioned that generally by creating those paths in AutoMod and describing them for each section a blocking system needs to be defined, which carries about the amount of vehicles that are allowed to be in that specified section of the path. Hence the capacity of the path restricts the processes and gives a deeper understanding of the overall system (Kuo and Huang 2006). However objects creation, especially paths definition for vehicles, requires resources.

Thus, both tools provide possibility to create a model, where restrictions or peculiar cases, related to the real system, are scripted. In fact AutoMod is able to specify the processes within the model description in more detailed and flexible way.

### Simulation

AutoMod and ED Log provide nearly the same simulation environment with similar functionalities, such as:

- Run control window
- Animation/simulation step
- Alarm, etc.

Both tools are equipped with good analyzing functions and provide various reports for any object in the created model at any time during simulation.

Providing an assumption for compared tools:

- Tools to run a simulation are almost the same in both applications, though AutoMod requires more resources, due to many objects inserted in a model
- Within the main objects and logic AutoMod is more flexible
- ED Log provides easy access and simple use of the working environment

So depending on the study case/logistics system different simulation tools can be chosen. Afterwards all the studies are presented to the client, who based on the information, can require each tool's vendor to create the exact model corresponding to its warehousing system. As soon as these models are created, the simulation tools can be studied once again, in order to choose the final tool. Accomplishing the model by the chosen simulation tool, all the theoretical ideas related to loads positioning and processes in the model, can be researched by running the simulation and the obtained results can be investigated for further actions.

## CONCLUSION

The investigated and described tools demonstrate that related to the defined task various tools can be chosen concerning what kind of task they pursue to solve. Moreover it is useful to keep in mind the questions that the model is tend to answer, else the model can be massive or contain information with so little detail making it not appropriate. The before mentioned model is more concerned about loads positioning and processes within the system. The aim of inventory positioning and management particularly relates to minimization of a travel distance in the warehouse. Corresponding changes in inventory positioning will affect and may optimize the processes. Theoretical views and shortcuts can be pre-checked by the simulation model.

## REFERENCES

Frazelle, E. 2002. *World-Class Warehousing and Material Handling*, McGraw-Hill, New York; London.

Merkuryev, Y., G. Merkuryeva, M.A. Piera and A. Guasch 2009. *Simulation-Based Case Studies in Logistic: Education and Applied Research.* Springer-Verlag, New York.

Merkuryeva, G, C.B. Machado, A. Burinskiene 2006. "Warehouse simulation environments for analyzing order picking process". In *Proceedings international mediterranean modeling multiconference*, 475-480.

Yuriy, G. and N. Vayenas. 2008. "Discrete-event simulation of mine equipment systems combined with a reliability assessment model based on genetic algorithms." *International Journal of Mining, Reclamation and Environment* 22, No.1 (Mar), 70-83.

Kuo, C.-H. and C.-S. Huang. 2006. "Dispatching of overhead hoist vehicles in a fab intrabay using a multimission-oriented controller." *International Journal of Advanced Manufacturing Technology* 27, No.7/8 (Jan), 824-832.

**CHRISTIAN-ANDREAS SCHUMANN**, born 1957 in Chemnitz (Germany), studied Industrial Engineering at the 'Chemnitz University of Technology' (CUT), doing his first doctor's degree in 1984 and second doctor's degree in 1987. He was appointed associate professor for plant planning and information processes at CUT in 1988. In 1994 he became professor for business and engineering information systems at the 'University of Applied Sciences Zwickau'. Since March 2003 till June 2009 he was dean of the faculty 'Business and Management Sciences' at Zwickau. Currently he is also director of the 'Center for New Forms of Education' and director of the "Central German Academy of Further Education'.

**NOEMI NIKOGHOSYAN** was born in Yerevan, Armenia and went to the Yerevan State University, where she studied Informatics and Applied Mathematics. In 2008 she obtained her Masters degree in the field of Information Systems Management. Since March 2009 she attends the University of Applied Sciences Zwickau for PhD studies.

# A FOUR HUNDRED VARIABLE NONLINEAR TRANSPORTATION PROBLEM

William Conley
Department of Business Administration and Mathematics
University of Wisconsin at Green Bay
Green Bay, Wisconsin 54311-7001
U.S.A.
E-mail: Conleyw@uwgb.edu

## KEYWORDS

Transportation, nonlinear multi stage Monte Carlo optimization.

## ABSTRACT

The standard mathematical transportation problem is one that attempts to deliver a product in quantity from k locations to r destinations where the k times r unit shipping costs are all linear. The linearity assumptions make the problem much easier to solve mathematically with the simplex algorithms or other modifications of the theory of linear systems of equations. However, in the real world of global economics and big business, there may be substantial discounts for shipping in quantity, causing the k times r unit shipping costs to be nonlinear. Therefore, presented here will be a 400 variable nonlinear transportation problem involving the quantity shipment of a product from the 20 factories that produced it to 20 warehouses that store it, before further shipments or sales. The multi stage Monte Carlo optimization (MSMCO) solution technique will be used to solve it.

## THE FOUR HUNDRED VARIABLE PROBLEM

A global company is producing a product in bulk at 20 factory locations and desires to ship it to its 20 warehouses at minimum cost. A textbook like (Mizrahi and Sullivan, 1993) details several examples of the classic linear transportation problem of this type. However, management decided to use the nonlinear simulation (MSMCO) approach in (Conley, 2007), where a 100 variable problem was solved, to see if it will work on a larger problem with a nonlinear cost structure.

Specifically, the company is producing 12,210,000 units of the product at its 20 factories in amounts of 601000, 602000, 603000 … 620000. The demands at its warehouses are for 620000, 619000, 618000 … 601000 units. Therefore, supply exactly equals demand. If they are not equal less than or greater than constraints could be substituted for some of the equations. The company would like to drive the total shipping cost down to 3000000 Euros or less. Therefore, the total shipping cost equation

$$C = \sum_{i=1}^{20} \sum_{j=1}^{20} (.049* (i+j)*X(i,j)**(.59+.01(i+j)))$$ is set equal to 3000000. Note* is multiply and ** is raise to a power and X(i,j) are the four hundred amounts leaving the 20 factories bound for the 20 warehouses. Additionally, there are 40 linear equations each with the right 20 X(i.j) variables added up and equal to the 40 supply and demand values (601000 … 620000) at the factories and warehouses. Then the multi stage Monte Carlo optimization technique attempts to minimize the sum of the absolute values of the differences between the left and right hand side of this 41 equation 400 variable system. Note that because the cost equation is larger than the other 40 equations, its absolute value of the difference between its left and right hand side is multiplied by .20 to yield a smoother simulation.

Forty thousand sample solutions are looked at in each of the fifty stages in an ever decreasing size and moving search of the four hundred variable feasible solution space, which took several minutes on a desk top PC.

Table 1 shows the 50 stage errors. Stage one has an error of 207 million. The stage two error is 196 million and the stage three error is a not so good 187 million plus. However, by stage 27, the total error is down to 100,307. The stage 50 value is 95,820 which actually is an answer that solved all 40 supply and demand equations and drove the total cost well below 3,000,000 Euros.

Table 1: The Fifty Stage Errors

| | |
|---|---|
| 1 | 207069728.0000 |
| 2 | 196366688.0000 |
| 3 | 187395616.0000 |
| 4 | 174793200.0000 |
| 5 | 158638736.0000 |
| 6 | 143646112.0000 |
| 7 | 121355216.0000 |
| 8 | 96001256.0000 |
| 9 | 79309168.0000 |
| 10 | 56039708 .0000 |
| 11 | 38526396.0000 |
| 12 | 25453022.0000 |
| 13 | 15016067.0000 |
| 14 | 7193102.5000 |

```
15 2756165.5000
16 1327842.3750
17 635305.5000
18 384795.0938
19 350901.0625
20 274989.0313
21 195820.3125
22 149414.0313
23 119703.4297
24 109956.1016
25 107745.1172
26 102534.0156
27 100307.8281
28 99567.8672
29 98093.1484
30 97440.7031
31 97008.5234
32 96601.0625
33 96551.8906
34 96321.5234
35 96061.2422
36 96001.9141
37 95970.5000
38 95924.5234
39 95905.1250
40 95876.2500
41 95853.3125
42 95839.2656
43 95834.8906
44 95829.6172
45 95826.2266
46 95823.3516
47 95822.7500
48 95821.1875
49 95820.8984
50 95820.1172
```

The 100 lines of the printout after the 50 stage error values give the 400 amounts (Table 2) to be shipped from each factory to each warehouse. This is followed by the 41 individual equation errors in the left hand column and their individual right hand side values that the solution is trying to meet (Table 3).

Table 2: The 400 Amounts to be Shipped

| | | | | |
|---|---|---|---|---|
| 1 | 31926.959 | 4214.806 | 18872.916 | 103858.344 |
| 1 | 6185.386 | 53068.820 | 6382.008 | 9141.631 |
| 1 | 4922.541 | 53018.273 | 15618.719 | 12086.894 |
| 1 | 39424.543 | 26946.668 | 5354.870 | 24193.111 |
| 1 | 168180.906 | 35320.289 | 48.697 | 12 32.811 |
| 2 | 7413.938 | 105835.609 | 8372.970 | 7315.602 |
| 2 | 1311.871 | 12278.799 | 36340. 574 | 115956.133 |
| 2 | 6333.243 | 21796. 154 | 3827. 702 | 8586.190 |
| 2 | 15767.674 | 14780.226 | 6715.897 | 7549.854 |
| 2 | 13176.757 | 214410.984 | 4623.430 | 6606.352 |
| 3 | 9172.682 | 9940.838 | 46636. 945 | 2377.955 |
| 3 | 27564.545 | 14270.757 | 21642.938 | 120. 291 |
| 3 | 83634.750 | 48382.887 | 5227. 291 | 3358.865 |
| 3 | 25255.357 | 7556.108 | 3539.321 | 11465.019 |
| 3 | 3189.121 | 36583.020 | 17150.098 | 240931.156 |
| 4 | 13832.025 | 11663.687 | 18796.586 | 29793.957 |
| 4 | 32367.961 | 33805.254 | 7620.573 | 22453.648 |
| 4 | 11909.007 | 99335.055 | 44986.246 | 9705.501 |
| 4 | 37465.863 | 24472.104 | 137169.703 | 25659.355 |
| 4 | 22883.514 | 10106.226 | 19086.414 | 3887.510 |
| 5 | 13397.643 | 4067.442 | 4008.233 | 27302.396 |
| 5 | 9868.265 | 44629.238 | 16133.400 | 36.242 |
| 5 | 122801.383 | 61558.176 | 58989.016 | 34255.504 |
| 5 | 8173.894 | 18210.809 | 44065.699 | 3362.846 |
| 5 | 31977.277 | 8521.550 | 18125.297 | 86515.789 |
| 6 | 982.707 | 12248.673 | 3532.055 | 1133.192 |
| 6 | 258.022 | 73489.570 | 1132.997 | 263751.313 |
| 6 | 31.612 | 7505.322 | 765.589 | 22616.918 |
| 6 | 553.428 | 144.507 | 1189.255 | 11570.354 |
| 6 | 16527.203 | 977.956 | 151665.969 | 44923.344 |
| 7 | 36038.074 | 89.862 | 1885.216 | 20538.830 |
| 7 | 27762.887 | 44049.625 | 3576.303 | 28320.230 |
| 7 | 17169.197 | 69556.344 | 1805.097 | 55765.168 |
| 7 | 49377.555 | 53007.797 | 8599.527 | 96442.109 |
| 7 | 29872.775 | 4240.161 | 62182.605 | 3720.633 |
| 8 | 8773.220 | 2698.171 | 7920.546 | 13626.102 |
| 8 | 32889.281 | 2511.973 | 13909.504 | 2054.495 |
| 8 | 4648.203 | 24687.941 | 50347.730 | 90686.227 |
| 8 | 29659.045 | 145306.250 | 62368.234 | 60499.574 |
| 8 | 20872.580 | 5329.387 | 25627.373 | 8584.267 |
| 9 | 181344.438 | 93231.680 | 4244.041 | 13000.969 |
| 9 | 24560.098 | 16944.158 | 18629.488 | 9504.710 |
| 9 | 501.126 | 3001.963 | 1723.916 | 12780.591 |
| 9 | 21607.453 | 5939.295 | 102040.383 | 29537.207 |
| 9 | 17355.246 | 20087.896 | 8416.941 | 27548.742 |
| 10 | 3711.300 | 17619.199 | 1856.572 | 14797.232 |
| 10 | 32499.875 | 52188.629 | 6776.218 | 19824.574 |
| 10 | 28922.313 | 9190.682 | 52233.410 | 47608.570 |
| 10 | 111214.305 | 46525.727 | 25602.506 | 34631.887 |
| 10 | 14907.270 | 12306.388 | 11218.389 | 67302.102 |
| 11 | 14248.367 | 76616.867 | 4406.004 | 26232.998 |
| 11 | 83049.164 | 18538.830 | 3014.750 | 71646.672 |
| 11 | 2659.584 | 3013.865 | 6873.692 | 1755.711 |
| 11 | 36786.820 | 67860.938 | 16037.059 | 39425.855 |
| 11 | 25163.887 | 24760.592 | 70667.719 | 17240.623 |
| 12 | 2299.139 | 83765.602 | 7976.785 | 20608.139 |
| 12 | 19800.127 | 39229.570 | 44482.672 | 826.966 |
| 12 | 123556.188 | 26847.818 | 63473.574 | 14073.945 |
| 12 | 42306.762 | 10102.052 | 9419.479 | 31160.682 |
| 12 | 17926.078 | 43854.691 | 2688.627 | 4601.403 |
| 13 | 1247.005 | 44793.348 | 12181.640 | 12949.977 |
| 13 | 10925.440 | 18508.965 | 113181.969 | 571.761 |
| 13 | 128955.516 | 4581.390 | 6055.940 | 76819.391 |
| 13 | 9294.729 | 12326.066 | 1540.196 | 80480.297 |
| 13 | 26339.068 | 11033.644 | 28973.080 | 7240.632 |
| 14 | 40094.496 | 13755.093 | 16674.852 | 24391.730 |
| 14 | 89097.805 | 21657.246 | 35883.516 | 3223.303 |
| 14 | 40464.898 | 12910.127 | 15722.387 | 39910.254 |
| 14 | 80952.539 | 1242682 | 10352.465 | 22344.951 |
| 14 | 21283.170 | 33782.461 | 72991.523 | 10264.585 |
| 15 | 2735.350 | 972.885 | 14485.790 | 58575.391 |
| 15 | 53951.402 | 2315.690 | 33325.590 | 11969.362 |
| 15 | 8534.486 | 28303.590 | 52802.191 | 14811.463 |

| | | | | |
|---|---|---|---|---|
| 15 | 21803.061 | 69159.805 | 129209.453 | 40061.457 |
| 15 | 27995.459 | 2976.901 | 12355.374 | 19655.387 |
| 16 | 5829.950 | 116866.250 | 24431.602 | 56123.645 |
| 16 | 19216.490 | 14621.565 | 84393.422 | 3031.511 |
| 16 | 5020.337 | 48786.324 | 3088.195 | 45896.152 |
| 16 | 25419.039 | 48461.797 | 6235.414 | 24303.945 |
| 16 | 13935.256 | 16540.381 | 36398.305 | 6400.554 |
| 17 | 4878.172 | 638.883 | 303531.156 | 1384.108 |
| 17 | 5314.788 | 1758.665 | 130748.516 | 7123.229 |
| 17 | 4793.023 | 7044.114 | 1003.866 | 10292.043 |
| 17 | 3675.906 | 5839.354 | 2336.844 | 2353.484 |
| 17 | 30896.965 | 27930.344 | 45176.555 | 7280.006 |
| 18 | 4804.144 | 1121.812 | 17565.191 | 9102.415 |
| 18 | 3866.323 | 124345.258 | 14142.043 | 3764.096 |
| 18 | 4258.984 | 35547.379 | 209983.719 | 31810.904 |
| 18 | 4688.439 | 39075.465 | 24936.893 | 28753.842 |
| 18 | 7777.535 | 14068.345 | 11052.212 | 12335.534 |
| 19 | 216336.734 | 802.372 | 6104.568 | 65145.559 |
| 19 | 695.625 | 11264.538 | 6908.734 | 14745.004 |
| 19 | 8383.830 | 2300.562 | 2594.995 | 75172.859 |
| 19 | 47394.836 | 3152.571 | 7397.233 | 16798.533 |
| 19 | 34782.879 | 66392.430 | 10105.269 | 5520.938 |
| 20 | 1933.763 | 1057.027 | 79516.414 | 95741.594 |
| 20 | 123814.688 | 6522.832 | 8773.876 | 19934.926 |
| 20 | 1499.834 | 42632.055 | 13876.776 | 4007.203 |
| 20 | 2179.307 | 13889.871 | 10889.621 | 25405.686 |
| 20 | 71957.078 | 28776.330 | 10383.426 | 38207.699 |

Table 3: Equation Errors in Left Column

| | |
|---|---|
| 0.12500 | 601000. 00000 |
| 0.06250 | 602000.00000 |
| 0.06250 | 603000.00000 |
| 0.12500 | 604000.00000 |
| 0.00000 | 605000.00000 |
| 0.00000 | 606000.00000 |
| 0.06250 | 607000.00000 |
| 0.12500 | 608000.00000 |
| 0.00000 | 609000.00000 |
| 0.00000 | 610000.00000 |
| 0.00000 | 611000. 00000 |
| 0.37500 | 612000.00000 |
| 0.50000 | 613000. 00000 |
| 0.06250 | 614000.00000 |
| 0.00000 | 615000. 00000 |
| 0.06250 | 616000.00000 |
| 0.00000 | 617000. 00000 |
| 0.00000 | 618000. 00000 |
| 0.25000 | 619000.00000 |
| 0.06250 | 620000.00000 |
| 0.18750 | 620000.00000 |
| 0.00000 | 619000.00000 |
| 0.06250 | 618000.00000 |
| 0.25000 | 617000.00000 |
| 0.12500 | 616000.00000 |
| 0.00000 | 615000. 00000 |
| 0.00000 | 614000. 00000 |
| 0.06250 | 613000.00000 |
| 0.31250 | 612000.00000 |
| 0.18750 | 611000.00000 |
| 0.00000 | 610000. 00000 |
| 0.18750 | 609000.00000 |
| 0.06250 | 608000.00000 |
| 0.00000 | 607000. 00000 |
| 0.00000 | 606000.00000 |
| 0.12500 | 605000.00000 |
| 0.00000 | 604000. 00000 |
| 0.56250 | 603000.00000 |
| 0.00000 | 602000 . 00000 |
| 0.06250 | 601000.00000 |
| 95816.05469 | 3000000.00000 |

Note that the 40 supply and demand equations all have errors of less than one unit. The cost equation has an apparent error of 95,816 which is actually good new for the company. That is because the 5 (95816.055) = 479,080.275, is how much below 3,000,000 Euros the solution is. Therefore the company can meet its supply and demand requirements when shipping the 12,210,000 units at cost a cost of 3000000 – 479080.275 = 2,520,920 Euros. Remember that the cost equation error was multiplied by 5 because it had been artificially reduced by a factor of 5 (times 1/5) to have a smoother simulation.

## FURTHER EXPLANATION OF THE TABLES

Table 2 represents fifty random searches or Monte Carlo solution attempts where at each of the fifty Monte Carlo solution stages, 40,000 sample sets of 400 variable values are looked at, and the one with the smallest cost equation value is stored.

Then centered about this best answer so far (207069728), in stage two 40,000 more sample answers are looked at in a reduced region and its best answer is stored (196366668). This process is done fifty ever improving times, finally getting a very good answer, even less than the goal of 3,000,000 Euros.

Table 2 is the answer of how much to ship from each factory to each warehouse. Therefore, for example the 31926.959 units is how much is shipped from factory one to warehouse one. The value 4214.806 is how many units are shipped from factory one to warehouse two and so on for those first 20 values at the top of Table 2.

The next twenty values down in Table 2 are how much to ship from factory 2 to warehouses ;1, 2, … 20. Finally, down to the last five lines, which show how much to ship from factory 20 to each of the warehouses. The value in the lower right hand corner of Table 2 (38,207.699) is how much is shipped from factory 20 to warehouse 20.

The first 20 lines of Table 3 are the amounts produced in factories 1 through 20 (in order) on the right hand side.

The left hand side is the equation errors in our solution. The next 20 lines are the amounts the warehouses need (on the right) and equation errors from our solution on the left. The 41st line deals with the cost equation which was reduced below goal of 3,000,000 Euros.

## CONCLUSION

The multi stage Monte Carlo optimization (MSMCO) technique was used to solve a 41 equation 400 variable nonlinear system of equations in a goal programming setting where a total shipping cost of less than 3 million Euros was desired along with meeting the 40 supply and demand equation values. The best answer of seven solution tries was presented here. Each try of 2 million sample answers (50 stages of 40000 sample answers) took about four to five minutes of computer run time on an inexpensive desk top PC. Multi stage Monte Carlo optimization also was used by (Wong, 1996) on a different system of equations taken from applied mathematics to show its generality of use. Desk top computers are powerful enough to tackle fairly large nonlinear multivariate optimization problems..

## REFERENCES

Conley, W. C. 2007. "A Nonlinear Transportation Problem with One Hundred Variables." In Proceedings of 2007 European Simulation and Modeling Conference ESM2007 (St. Julians, Malta Oct 22-24) EUROSIS-ET1, Belgium 287-290.

Mizrahi, A. and Sullivan, M. 1993. *Mathematics for Business Life Science and Social Sciences*, 5th edition. John Wiley and Sons, New York.

Wong, J.Y. 1996. "A Note on Optimization in Integers," International Journal of Mathematical Education in Science and Technology, Vol. 27, No. 6, 865-874.

## BIOGRAPHY

**WILLIAM CONLEY** received a B.A. in mathematics (with honors) from Albion College in 1970, an M.A. in mathematics from Western Michigan University in 1971, an M.Sc. in statistics in 1973 and a Ph.D. in mathematics - computer statistics from the University of Windsor in 1976. He has taught mathematics, statistics, and computer programming in universities for over 30 years. He is currently a professor of Business Administration and Statistics at the University of Wisconsin at Green Bay. The developer of multi stage Monte Carlo optimization and the CTSP multivariate correlation statistics, he is the author of five books and 200 publications world wide. He is a member of the American Chemical Society, a fellow in the Institution of Electronic and Telecommunication Engineers and a senior member of the Society for Computer Simulation.

# MANUFACTURING MANAGEMENT TOOLS

# OPTIMUM ALLOCATION OF INSPECTION EFFORT IN MULTISTAGE MANUFACTURING PROCESSES

Ali G. Shetwan
Valentin I. Vitanov
Design Manufacture and Management,
School of Engineering, Durham University
South Road, DH1 3LE, UK.
E-mail: a.g.shetwan@durham.ac.uk v.i.vitanov@durham.ac.uk

**KEYWORDS**
Manufacturing, Quality, Resources Allocation, Inspection Optimisation.

## ABSTRACT

The allocation of inspection effort (AIE) in multi-stage manufacturing system has been studied extensively over the last fifty years. The objective of this paper is to review the existing approaches, propose a classification of the available models in terms of the type of manufacturing system that they refer to and the applied solution methods and examine the effectiveness of the inspection strategies by developing appropriate generalised algorithm and software tool. The review revealed firstly that inspection allocation problem has been studied comprehensively by using variety of analytical and Monte-Carlo simulation methods rather than combination of both simulation techniques in a simulation-optimisation framework. Secondly large proportion of the papers focuses on several work stations representing part of a manufacturing line without attempting to solve the global optimisation problem which lead to solutions based on complete enumeration that are known to be computationally ineffective when the number of workstations increase. The developed simulation program demonstrated that methods determining the position of inspection by using complete enumeration method (EM) are of limited use in the majority of manufacturing situations when the number of workstations exceeds eighteen. This led to the development of a heuristic algorithm the performance of which was compared with the complete enumeration algorithm. It was found that heuristic method can derive an acceptable solution significantly faster. At present authors continue to develop heuristic algorithms for the AIE problem and methaheuristics using biologically inspired techniques.

## INTRODUCTION

The quality management policies in the majority of companies evolve continuously over a number of years by focusing on quality issues that are critical at any given instant of time. This approach usually focuses on particular critical operations and does not take into account the need for global analysis of a manufacturing system quality problem. As a result of that quality policies do not utilise fully the available financial, human and equipment resources. In the same time, present economic environment, the reduction of waste becomes of paramount importance because the increase in the product cost affects the overall competitiveness of the manufactured products. This paper targets specifically wastes resulting from unidentified defective items being processed unnecessarily during following manufacturing operations. The solution of inspection effort allocation issues needs to adopt the corresponding utility strategies. Such strategies aim to allocate an economically appropriate level of inspection effort by striking a balance among the different cost components connected with inspection, scrap, repair and replacement due to quality failure, and/or the warranty penalty in the case where a nonconforming product has been shipped to customers. Inspection-oriented strategy focuses on optimization that minimizes the expected total manufacturing cost, maximizes the quality and is capable of delivering the demanded quantities of the product. The expected total cost consists of the manufacturing cost, inspection cost, internal failure cost and external failure cost. The general problem of allocation inspection effort in manufacturing systems can be divided into two sub-problem categories aiming to answer the following research questions: (1) If the requested quality level, production volume and product costs are fixed what is the optimal number of inspection stations for a manufacturing process? (2) If the number of inspection stations is fixed for a particular process where the optimal places to position are them in order to optimise the quality, cost, and delivery manufacturing attributes?

Vast majority of the reviewed papers focused on the second question of optimal allocation of inspection station. The objective of this paper is to present an overview on how these questions have been answered and to examine the inspection strategies by developing the appropriate algorithm and software tool.

## INSPECTION ALLOCATION PROBLEM

### Problem background

The presented paper attempts to cover the major sources for the last fifty years that were available to the authors. The procedure of making decisions of whether or not to inspect a final or semi-finished product at every processing workstation is shown schematically in Figure 1. It is assumed that if inspection is performed after every workstation, the scrap and rework costs will stay at a minimum level. These savings have to be considered against the inspection costs which include equipment, staff, time, shop floor space and increase the number of works in process. Therefore, if these in process inspections are performed too often unnecessary costs will occur. In a practical situation usually the expectation is that manufacturing processes at each work station are capable of achieving the required quality tolerances for 99.97% of the items which is the 3σ level when the process is on target. It is possible that a shift of up to 1.5σ from the target quality level may occur undetected subject to the quality procedure which may lead to drop in quality level down to 93.3% and when such situation occurs our inspection stations have to be positioned in a way that will minimise the overall lost of profit. If the shift is larger than that we consider that the problem is not any longer a quality monitoring but a manufacturing issue that has to be fixed by different means. The purpose of inspection allocation strategy is to allocate an economically appropriate level of inspection activity by determining the correct balance among different cost components indicated above.

### Structure of the Line

Products are often processed through multi-stage production systems, where incoming material is transformed into the finished product in a chain of different processing stages. There are several types of production systems such as: (i) serial systems; (ii) assembly systems; and (iii) non-serial systems. In a serial production system, the incoming material passes through sequence of processing workstation to the final product. See (Shiau 2002) and (VanVolsem et al. 2007). Whereas in an assembly manufacturing system, at a certain stage the product may be fixed or assembled with products from other processing lines, see (Penn and Raviv 2007). A system that is neither serial nor non assembly falls in non-serial system. See (Taneja and Viswanadham 1994) and (Narahari and Khan 1996). However, it is more difficult to determine undetected defects in assembly system than in serial line. The difficulty arises due to assembly stages at which multiple serial lines join to form a single serial line. At such assembly stages, the number of undetected defective output flow items of the assembly stage entering the assembly line stage depends on the proportion of defective items leaving all series lines to that assembly stage.

### Inspection Time

In the reviewed papers, inspection time plays a major role to the total manufacturing costs. Longer inspection times strain the inspection capacity which may cause increased inspection errors. Saxena et al. (1990) explained this by using a simulation model to examine the performance of five inspection station allocation heuristics on the basis of job completion time in serial production systems under different operating conditions. They found that inspection time was the most influential factor for the selection of a particular heuristic rule.

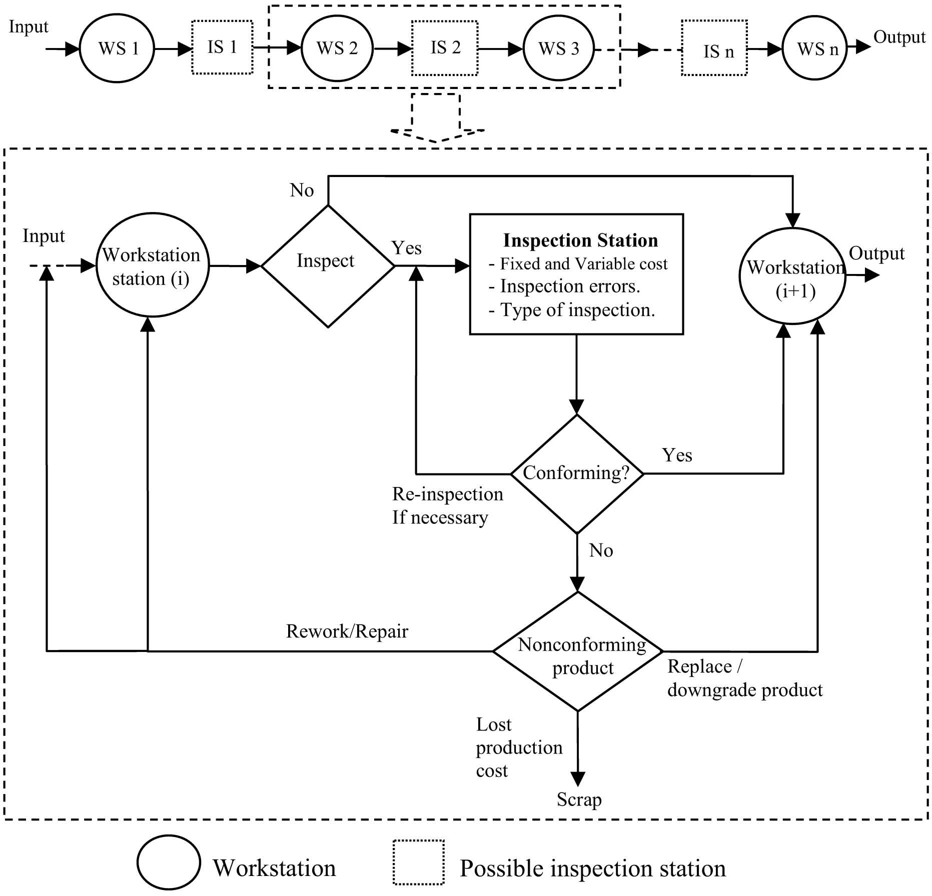


Figure 1: Inspection Allocation Problem in Multistage Manufacturing Processes

The vast majority of the papers were assumed that the inspection time for each inspection station can be represented by the inspection cost. Indeed only one paper in the optimisation techniques (Lee and Unnikrishnan 1998) was used inspection time as constraint to the objective function of the total cost. They found that by considering inspection time as constraint, the number of inspection plans can be reduced.

### Repair of Defects

As described in Figure 1 once defective item is detected to be non-conformed to specifications during inspection, certain actions will be taken to repair, replace, or simply scrap it. However, defective product, when allowed to pass through the production stream, might cause, time delaying and become costly to repair at a subsequent stage of operation. To account for this in a model some researchers have assigned a reparability level for defects. For example (Lee and Unnikrishnan 1998) and (Shiau 2003a) have assumed deterministic assignment of reparability and (Narahari and Khan 1996) and (Barad1990) have adopted a probabilistic approach which assumes that a defect is repairable with a given probability. The repair occurs only when every nonconforming predefined quality of a product is larger than specification limit. In the absence of both repairable and replacement, the production volume in the model will shrink as a result of inspection. In real life without replacement or repair larger lot sizes have to be introduced in order to meet production plans and to avoid delivery delays.

### Nonconforming Products

Products may become nonconforming because of improper performance of a processing operation. The chance that a unit will become nonconforming at a given stage is referred to as the nonconforming processing rate for the stage, and may be constant or variable, and may alternate between an acceptable level and an out-of- control level. A given processing stage may cause a single type of nonconformity or multiple types. In the allocation inspection station, the majority of the reviewed papers have assumed that, each workstation may have a specific probability of producing defective parts. Products considered being nonconforming and subsequently removed from the production flow may have some or no salvage value. The salvage value represents the revenue generated by selling the rejected items as scrap or lower grade products see (Eppen and Hurst model 1974). They assumed that a unit rejected by inspection, whether good or defective, is always removed incurring a salvage cost which might be negative.

### Inspection Capability

During the inspection operation two types of errors may be generated by the inspection procedure, type-I error and type- II error, see Figure 1. The type-I error refers to a rejection a good items and is also known as producer risk, whereas type- II error refers to acceptance of a nonconformce items and is also known as consumer risk. A type II error is usually more serious. Not all the authors have considered both types of errors. (Rebello et al. 1995) and (Tannock 1997) have only considered one of the two types and some other authors have simply assumed a perfect inspection (Narahari and Khan 1996) and (Penn and Raviv 2008).

## MODELLING FEATURES

An inspection quality allocation plan is usually solved through an optimization formulation. In this section, it types of cost components will be discussed followed by the solution approach proposed by the researchers. Table 1 shows a summary of the classifications where each publication is represented by the first author's name followed by a two-digit publication year in order to conserve space.

### Cost Components

The manufacturing cost of a product is one of the major factors under consideration. The usual requirement is that the products to be produced at an acceptable quality level, and minimum manufacturing costs. Perhaps for that reason, the majority of the researchers chose to focus on specific cost components related to quality failures (internal failure cost and external failure cost). The internal failure costs occur inside the company, such as the costs of reworking, scrapping and replacement. The external failure costs occur after the goods are shipped to customers, such as the costs of replacement, repairing, and quality loss. Vast majority of the reviewed papers have not considered in their models all items of external failure cost. They just represented them as aggregated (penalty cost). The penalty cost is usually associated with final production of undetected nonconforming items that reach at the customer. Some other researchers did not consider at all any item of external failure cost see (Tannock and Saelem 2007) and (Barad 1990). Whereas a few papers are considered the most other items of external failure cost in their models such as quality loss, replacement cost and repair cost see for example (Shiau 2002) and (Shiau 2007). The repair cost occurs only when every nonconforming predefined quality of a product is larger than specification limit. Otherwise, the replacement cost will take place.

Other costs included are inspection cost and manufacturing cost. Inspection cost occurs only when an inspection station is located after workstation; otherwise manufacturing cost will take place. Inspection cost is a sum of the fixed cost and the variable cost. The fixed cost is a sum of the costs connected with test-equipment installation, setup, etc. The variable cost is the total number of conforming parts and the number of defective parts produced at inspection station multiplied by the inspection cost for carrying out of 100 % item-by-item.

### Solution Approaches

In the inspection allocation problems, the most common treatment that the models are developed with objective of minimising the expected total cost per unit produced. The total cost includes some or all of the following costs, internal failure cost, and external failure cost, inspection cost, and manufacturing cost. Table 1 shows these costs

Table 1 Classification of Models According to System Features

| System Features | Categorization | Publications |
|---|---|---|
| Production Line | Serial | Lindsay (64), Eppen (74), Shiau (02), Lee (98), Shiau (07), Barad (90), Raz (00), VanVolsem (07), Shaiu (03a), Shaiu (03b), Rebello (95), Tayi (88), Raghavachari (91), Rau (05), Jewkes (95), Bai (96), Ballou (85), Chengalur ( 92), Taneja (94), Freiesleben (06), Kakade (04), Raz (91), Langner (02), Galante (07), Chen (98), Yao (99), Kim (08), Jang (02), Taneja (94) Tannock (95), Tannock (97), Lee (96), Siemiatkowski (06), Neu (02), Tannock (07), Saxena (90) |
| | Assembly | Penn (08), Hadjinicola (03), Penn(07), Chen (99), Valenzuela (04), Gardner (95), Clark (99), Estrop (92) |
| | Non-serial | Taneja (94), Narahari (96), Rau (05) |
| Constraints | Inspection time | Lee (98) |
| | Limited inspection station | Lee (98), Shaiu (03a), Shaiu (03b), Shiau (02), Bai (96), Ballou (85), Chengalur ( 92), Penn (08), Penn (07),Taneja (94), Shiau (07), Jang (02) |
| | AOQL | Lindsay (64), Taneja (94) |
| | Limited budget | Rebello (95), Hadjinicola (03) |
| | Rate of inspection | Kakade (04) |
| Inspection Capability | Type I and II error | Eppen (74), Lee (98), Shaiu (03a), Shaiu (03b), Shiau (02), Rau (05), Bai (96), Ballou (85), Chengalur ( 92), Raz (00), Taneja (94), Shiau (07), Raz (91), Langner (02), Galante (07), Clark (99), Gardner (95) |
| | Type II error | Rebello (95), Tannock (97) |
| | Free of error | Tayi (88), Raghavachari (91), Barad (90), VanVolsem (07), Jewkes (95), Hadjinicola (03), Narahari (96), Lindsay (64), Penn (08), Penn (07), Freiesleben (06), Chen (98), Kakade (04), Chen (99), Yao (99), Kim (08), Jang (02), Valenzuela (04), Saxena (90), Tannock (95), Lee (96), Siemiatkowski (06), Neu (02), Tannock (07), Estrop (92) |

Table 1 Classification of Models According to System Features (Continued)

| System Features | Categorisation | Publications |
|---|---|---|
| Internal Failure Cost | Rework/Repair | Eppen (74), Lee (98), Shaiu (03a), Shiau (02), Rebello (95), Rau (05), Barad (90), VanVolsem (07), Jewkes (95), Hadjinicola (03), Narahari (96), Bai (96), Raz (00),Taneja (94), Shiau (07), Freiesleben (06), Kakade (04), Raz (91), Langner (02), Galante (07), Chen (99), Chen (98), Yao (99), Kim (08), Jang (02), Valenzuela (04), Tannock (95), Tannock (97), Neu (02), Tannock (07), Saxena (90), Clark (99) |
| | Replace | Shaiu (03a), Shaiu (03b), Shiau (02), Rebello (95), Barad (90), VanVolsem (07), Clark (99) |
| | Scrap | Eppen (74), Lee (98), Shaiu (03a), Shaiu (03b), Shiau (02), Rebello (95), Tayi (88), Raghavachari (91), Rau (05), Barad (90), Hadjinicola (03), Narahari (96), Lindsay (64), Chengalur ( 92), Raz (00), Taneja (94), Shiau (07), Freiesleben (06), Raz (91),Langner (02), Galante (07), Chen (99), Kim (08), Tannock (07), Neu (02), Gardner (95), Clark (99), Lee (96), Tannock (95), Tannock (97), Siemiatkowski (06), Saxena (90), Estrop (92) |
| External Failure Cost | Replacement | Lee (98), Shiau (07), Shiau (02), Shaiu (03b) |
| | Repair | Lee (98), Shiau (07), Shiau (02), Shaiu (03b) |
| | Quality loss | Shiau (07), Shiau (02), Shaiu (03b) |
| | Penalty | Penn(07), Raz (00), VanVolsem (07), Rau (05), Rebello (95), Tayi (88), Raghavachari (91), Rau (05), Jewkes (95), Bai (96), Ballou (85), Penn (07), Chengalur ( 92), Penn (08), Raz (00), Taneja (94), Kakade (04), Raz (91), Galante (07), Chen (98), Yao (99), Valenzuela (04) |
| Inspection Cost | Fixed | Tayi (88), Ballou (85), Chengalur ( 92), Raz (00), Chen (99), Jang (02), Tannock (07), Neu (02), Gardner (95), Clark (99), Estrop (92) |
| | Variable | Rebello (95), Rau (05), Barad (90), VanVolsem (07), Jewkes (95), Hadjinicola (03), Lindsay (64), Taneja (94), Shiau (07), Kakade (04), Raz (91), Langner (02), Galante (07), Chen (98), Yao (99), Valenzuela (04), Lee (96), Tannock (95), Tannock (97), Siemiatkowski (06) |
| | Fixed and Variable | Raghavachari (91), Bai (96), Penn(07), Penn (08), Freiesleben (06), Saxena (90) |
| Manufacturing Cost | | Tayi (88), Raghavachari (91), Barad (90), Hadjinicola (03), Penn (08), Ballou (85), Chengalur ( 92), Penn (07), Taneja (94), Shiau (07), Freiesleben (06), Langner (002), Galante (07), Chen (99), Kim (08), Jang (02), Valenzuela (04) Clark (99), Gardner (95), Saxena (90), Lee (96), Siemiatkowski (06), Tannock (07), Estrop (92) |

regarding to each paper. However, not all the papers try to minimise the total cost. As shown from Table 2, a few papers have decided to maximise the production capacity, see for example (Rebello et al. 1995) and (Valenzuela et al. 2004). That usually occurs when an inspection scheduling problem and the allocation problem is concurrently considered (Mandroli et al. 2006). Constraints that were used by the researchers in the optimization of an inspection are mostly related to the characteristics of the manufacturing system such as the structure of the system, the type of defect and the type of inspection. The most researchers derive nonlinear programming problem for their total cost functions, because of the nature of the inspection allocation problem function in which some of the decision variables can only have integer values for example whether or not to inspect at workstation and the serial number of the inspection stations. Dynamic programming (DP) approach also has been studied extensively because of the multistage arrangement of a manufacturing system that is described well by stages and states of the DP models (Mandroli et al. 2006).

Some previous publications present an interesting remark. For example (Lee and Unnikrishnan 1998) and (Shiau 2002, 2007) have pointed out that the DP approach employed in previous methodologies becomes quite impractical as the set of possible combinations grows exponentially. However, they do not provide any material evidence to prove their remark. Other techniques included genetic algorithm and simulated annealing, see Table 2. In order to simplify their model, the majority of simulation papers considered in this survey focus on simple process. For example a single process issues was investigated by (Clark and Tannock 1999) and (Estrop et al. 1992). Most of the simulation papers attempted to answer the second of the research questions. They examined the performance of inspection station allocation through heuristics rules on the basis of the parameters considered in serial production systems under different operating conditions. See for example (Saxena et al. 1990), (Lee and Chen 1996), (Siemiatkowski and Przybylski 2006) and (Gardner et al. 1995). Proposed simulation scenarios examine the allocation and sequencing of the inspection operations with regard to the incurred costs of quality assurance and the requirements of efficient system operation.

The review has concluded that the following mathematical methods were used out of 44 papers 20% dynamic programming, 25% nonlinear programming, 11% genetic algorithm, 7% branch and bound and Markov decision, 4% simulated annealing, and 2% Tabu search and linear programming. Also the review has shown that 23% out of 44 papers have used Monte-Carlo simulation technique.

## DEVELOPMENT OF MODEL

### Model Description

To examine the inspection allocation problem, serial multistage manufacturing system has been studied. Figure 1 illustrated the characteristics of the type of multistage system under consideration as follows:

1. The system is considered to be made up of 10 workstations arranged serially and parts are entering the system in batches.
2. Each workstation has a specific probability of producing defective parts.
3. A 100% inspection screen is applied to all parts processed in workstation if an inspection station is performed after it in the sequence.
4. Only one final product is considered in the system.

Table 2 Classification of Models According to Solution Approach

| System Features | | Categorisation | Publication |
|---|---|---|---|
| Solution Approach | Optimisation | Dynamic Programming | Eppen (74), Lindsay (64), Bai (96), Penn (08), Raz (00),Yao (99), Chen (98), Chengalur (92), Raghavachari (91) |
| | | Simulated Annealing | Chen (99), Kakade (04), |
| | | Genetic Algorithm | Galante (07), Freiesleben (06), Shiau (07), Taneja (94),VanVolsem (07) |
| | | Tabu Search | Valenzuela (04) |
| | | Branch and Bound | Penn (07), Langner (02), Raz (91) |
| | | Nonlinear Programming | Ballou (85), Narahari (96), Jewkes (95), Lee (98), Hadjinicola (03), Shaiu (03a), Shaiu (03b), Shiau (02), Barad (90), Rau et al. (05), Rau (05) |
| | | Linear Programming | Rebello (95) |
| | | Markov Decision | Jang (02), Kim (08), Tayi (88) |
| | Monte-Carlo Simulation Technique | | Tannock (95), Tannock (97), Lee (96), Siemiatkowski (06), Neu (02), Tannock (07), Saxena (90), Gardner (95), Clark (99) Estrop (92) |
| Objective Function | | Minimum Total Cost | Eppen (74), Lindsay (64), Shaiu (03a), Shaiu (03b), Shiau (02), Tayi (88), Raghavachari (91), Rau et al. (05), Rau (05), Barad (90), VanVolsem (07), Jewkes (95), Narahari (96), Bai (96), Ballou (85), Chengalur (92), Raz (00), Taneja (94), Shiau (07), Freiesleben (06), Kakade (04), Raz (91), Langner (02), Galante (07), Chen (99), Chen (98), Yao (99), Jang (02), Kim (08) |
| | | Maximum Profit | Rebello (95), Hadjinicola (03), Penn (08), Penn (07), Valenzuela (04) |

5. Two types of inspection errors are considered in the system. A type I error involves the classification of a conforming unit (CU) as a nonconforming unit (NCU), and a type II error means the classification of an NCU as a CU.
6. The system has limited number of inspection stations (e.g. five stations). Each inspection station can be assigned to perform inspection operation for one or more workstations.
7. Nonconforming items can either be scrapped or sent for rework. At each inspection station there is exists a specific probability of selecting nonconforming items for rework.
8. Rework items may be incurring defects in the subsequent reworking process.
9. The selection of an inspection station is subjected to a time constraint.

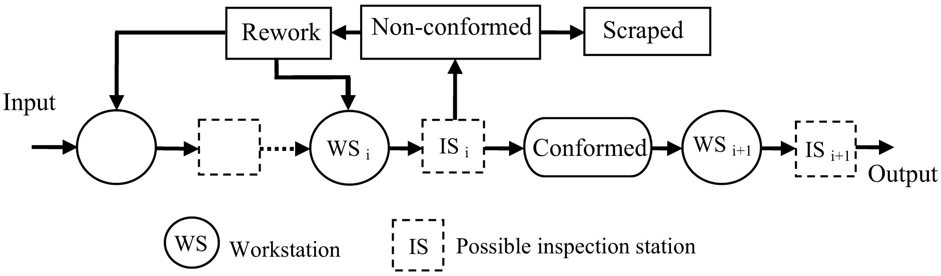


Figure 2: Serial Manufacturing Processes with Inspection Stations

**Description of Indices**

m: Refers to the inspection station assigned;
K: Refers to workstation.

**Description of Inputs**

n: Number of workstations in the system;
Q: Number of parts entering the system;
q: Number of inspection stations in the manufacturing system;
$\alpha_k$: Probability that the *m* inspection operation incorrectly classifies a conforming unit (CU) as a nonconforming unit (NCU);
$\beta_k$: Probability that the *m* inspection station incorrectly classifies a NCU as a CU;
$\Delta_m$: Probability of repairing a defective unit at the *m* inspection station;
$E_m$: Probability of repairing parts incurring a defect on subsequent processing at the *m* inspection station;
$D_{km}$: Direct cost of material to repair a defect part;
P: Final sale price of each unit is sold;
$Z_k$: Probability of a nonconforming of part processing at the *k* workstation;
Y: Direct cost of material to repair a defect at the customer's end;
$IT_m$: Processing inspection time of parts at *m* inspection station;
$g_k$: Multiplicative fraction of the manufacturing cost expressed as rework cost;
$x_k$: Unit scraping cost at the *k* workstation;
W: Percentage of parts replaced at the customer's end;
$I_{km}$: Unit inspection cost at *m* inspection station;
$fn_m$: Unit inspection processing time at *m* inspection station;
$V_k$ Unit manufacturing cost at the *k* workstation.

**Description of Variables**

$NG_k$: Number of conforming parts leaving the *k* workstation;
$ND_k$: Number of defective parts leaving the *k* workstation;
$RSC_{km}$: Rework scraped cost of parts on subsequent processing at *k* workstation;
$NR_{km}$: Number of parts sent back for rework to the previous workstation *k*
$MC_k$: Total manufacturing cost for parts at the *k* workstation;
$TIC_{km}$: Total inspection cost at *m* inspection station;
$NS_k$: Number of parts scraped at the *k* workstation;
$NRS_{km}$: Number of parts discarded after reworking processing;
$RC_{km}$: Rework cost at the k workstation;
TC: Total cost of manufacturing parts.

## Work Flow Analysis

Flow constraints consider the number of conforming parts departure a workstation or inspection location, and the number of defective parts entering a following workstation or inspection location. The following equation is represented the first stage. $NG_1 = Q(1-Z_1)$ (1)

For all other stations the equation is defined recursively as follows:

$$NG_k = [NG_{k-1} * (1-\alpha_{(k-1)m}) + ND_{k-1} * \beta_{(k-1)m} + NR_{k-1}] * (1-Z_k) \qquad (2)$$

The number of defective parts produced at first a workstation is: $ND_1 = Q * Z_1$ (3)

For all other stations, it is:

$$ND_k = [NG_{k-1} * \alpha_{(k-1)m} + (ND_{k-1} * (1-\beta_{(k-1)m})) + NR_{k-1}] * Z_k \qquad (4)$$

The number of parts classified as defective by inspection station *m* after workstation *k* but can be repaired is given by: $NR_k = NG_k * \alpha_{km} + ND_k (1 - \beta_{km})$ (5)

## COST MODEL ANALYSIS

### Expected Manufacturing Cost

This cost is assumed to be a sum of the material cost, overhead cost, and setup cost. The number of parts processed at station *k* is the sum of the number of parts correctly classified as conforming parts flowing into station *k* from the earlier process station and the number of defective parts incorrectly classified as conforming parts flowing into station *k* from the previous process station. Hence, the manufacturing cost ($MC_k$) is defined as follows:

$$MC_k = [NG_{k-1} * (1 - \alpha_{k-1}) + ND_{k-1} * \beta_{k-1}] * V_k \quad (6)$$

In the case where no inspection station is performed, $\alpha_{k-1} = 0$ and $\beta_{k-1} = 1$.

Thus: $MC_k = [NG_{k-1} + ND_{k-1}] * V_k$ (7)

### Expected Inspection Cost

Inspection cost is consists of sum of the fixed cost and variable cost. The fixed cost is sum of the costs connected with test-equipment installation, setup, calibration, etc. The variable cost is the total number of conforming parts and the number of defective parts produced at station *k* multiplied by the inspection cost for carrying out of 100 % item-by-item inspection of incoming batches. Therefore, the total inspection cost is given by:

$$TIC_{km} = FC_{km} + [(NG_k + ND_k)] * I_{km} \quad (8)$$

### Internal Failure Cost

The internal failure cost is the sum of reworking cost and scrap cost.
1. Reworking cost.
This is the cost of reworking a part identified as nonconforming at an inspection station. At each inspection station the nonconforming parts can be scrapped, sent back for repair or incorrectly classified as conformed parts. The number of parts as nonconforming but repairable are given by: $NR_{km} = [NG_k * \alpha_{km} + ND_k (1 - \beta_{km})] * \Delta_k$ (9)

Then the rework cost is: $RC_{km} = (NR_{km}) * (g_k * V_k)$ (10)

The rework parts may be incurred defects on subsequent processing as they did in the original process.

$$NRS_{km} = NR_{km} * E_k \quad (11)$$

2. Scraped cost.
This expression represents the number of non-repairable items produced at *k* station on detection subsequent *m* inspection stage.

$$NS_k = NG_k * \alpha_{km} + ND_k (1 - \beta_{km}) \quad (12)$$

The Scrap cost is: $SC_{km} = NS_k * x_k$ (13)

Also the scrap cost may result from a subsequent reworking process is given: $RSC_{km} = NRS_{km} * x_k$ (14)
Then the internal failure cost is given by:

$$IFC_{km} = SC_{km} + RSC_{km} + RC_{km} \quad (15)$$

### External Failure Cost

This is the cost incurred after the products have been sold to customers. The external failure cost (*EFC*), is the sum of the product of the number of defective parts replaced at the customer's end ($W * ND_k$), the sale price (*P*) of the part and the sum of the product of the number of defective parts repaired at the customer's end (1 - *W*) and the direct cost of materials to repair a defective unit (*Y*).

$$EFC = W * ND_k * P + (1 - W) * ND_k * Y \quad (16)$$

The inspection screen is applied to all parts processed in workstation if an inspection operation is performed, otherwise: $I_{km} = \alpha_k = \Delta_k = 0$ and $\beta_k = 1.0$
The total cost (*TC*) of processing and inspection of *Q* parts in an *n*-stage serial manufacturing system is given by the following equation:

$$TC = \sum_{k=1}^{n} (IFC_k + MC_k + IC_{km} + EFC) \quad (17)$$

The sum of the total cost of processing and inspecting the parts produced in the manufacturing system is expressed as the total system cost *TC*. The objective function for the inspection station allocation problem for a manufacturing system producing parts is expressed as follows:

$$Minimise \quad TC = \sum_{k=1}^{n} TC \quad (18)$$

In this paper, the objective function is constrained by time of inspection to reduce number of inspection plans and to maintain the nominal production rate. General time equation involves inspecting all parts plus set-up time at inspection station *m* in the manufacturing system. The constraints are the following:

$$IT_m \le [NG_k + ND_k] * fn_m + tss_m \quad (19)$$

Where: $IT_m \le T_m$

$$\sum_{k=1}^{n} q_k \le NI \quad (20)$$

Equation (20) shows that there are limited inspection stations available.

The allocation of quality inspection station problem grows exponentially with the number of workstations. For example, in a 21 workstations, there are more than 2,000,000 ($M=2^n$) ways to locate inspection positions and select a capable inspection station. Assume C consists of all assignment location combinations, then:

$$C = [(X_{11},\ldots X_{k1},\ldots X_{n1}),\ldots(X_{1m},\ldots X_{km},\ldots X_{nm}),\ldots (X_{1M},\ldots X_{km},\ldots X_{nM})]$$

If workstation *k* should be screened by an inspection station, then $X_{km}=1$, otherwise $X_{km}= 0$. A possible allocation plan among the lowest expected sum cost can then be determined after reconsidering every assignment location combination. As shown from Figure 3 that the processing time to solve the problem is exponentially grows with increasing number of workstation (WS). The experimental data was approximated using exponential regression model which has shown correlation r = 0.99 and the coefficient of the determination is $r^{2 =}$ 0.99 see equation (21).

$$\text{Time} = 0.0003e^{(0.87*\text{WS})} \qquad (21)$$

For example using equation (21) the duration of computation for 20 workstations is expected to be 180 hours. Therefore, it is impractical to allocate inspection places by using EM. In order to find solution when the number of workstation increases, a heuristic method was developed. It is introduced in the following section.

## HEURISTIC METHOD

The heuristic method (HM) aims to determine the inspection plan with the lowest total cost. The objective function of the total cost in this particular case is constrained by inspection time. The proposed method is named (Heuristic Method Time Constraint). It is referred further by the abbreviation (HMTC). In this case the number of items inspected is multiplied by inspection processing time allocated to each item which should be less than or equal to the inspection time assigned for the inspection station being considered, see (equation 19). This is done in order to maintain the nominal production rate and to reduce the number of feasible plans to be evaluated.

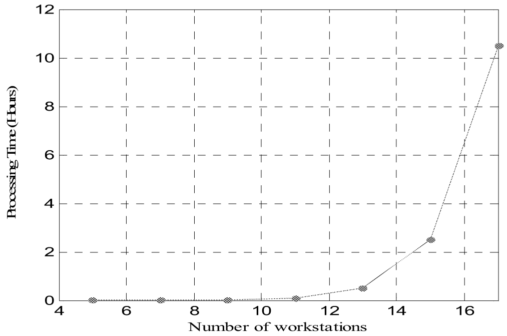


Figure 3: Schematic Showing the Duration of Computational Time in Relation to the Number of Workstations.

In the heuristic method, every assigned location in *C* is reviewed to determine the inspection plans that have to be assigned for one workstation. The inspection plan will be considered when $X_{km=}1$ and the sum of row elements of each of the generated inspection station plans match the number of inspection stations required. For example an inspection plan in the assignment location combination is (0111000110). It means that it has 5 inspection stations. Assume the available number of inspection stations is limited to 5. This inspection plan will be considered, because the number of inspection stations matches the number of inspection stations required. If the condition is not met, the inspection plan will be rejected, because it is unnecessary to check the inspection plans for unsatisfactory assignment location combination. The above approach will lead to avoiding unnecessary processing. Heuristic method idea is to minimise the cost by early identification of a non-conforming in the manufacturing processes. The following steps describe the heuristic procedure:

*Step*1 Generating the set of assignment location combinations *C*, for a multistage manufacturing system.

*Step*2 Based on limited inspection stations, decide number of inspection stations available.

*Step*3 Checking whether inspection plans matches the number of inspection stations available. If yes go to step 4, otherwise go to step 2.

*Step*4 Calculating the inspection time for the inspection station ($IT_m$) in the inspection plan. If ($IT_m$) less than or equal to the inspection time assigned for the inspection station being considered, go to step 5. Otherwise this inspection plan will not be considered. Go to step 3.

*Step*5 According to the order of workstations, if $X_{km}$=1 calculate total inspection cost, internal failure cost and manufacturing cost. If $X_{km}$ = 0, calculate total a manufacturing cost and external failure cost. Go to step 6.

*Step*6 Calculate the total inspection cost plan. Go to step7.

*Step*7 Checking whether all inspection stations are considered. If yes go to step 8, otherwise go to step 4.

*Step*8 Checking whether all inspection plans are considered. If yes go to step 9, otherwise go to step 3.

*Step*9 Determine the inspection plan that has the lowest cost.

## CASE STUDY AND DISCUSSION

The following simulation experiment was conducted when measuring the time used by the heuristic method to find the best positions for inspection stations. A multistage manufacturing system model with 10 workstations with different parameters arranged in a serial manner was used to allocate 5 inspection stations as shown in Table 3. The batch size used in the experiment was 100. The parameters were randomly generated using a uniform random number generator to evaluate processing time efficiency. It was found that the heuristic method (HM) and the one with time constraint (HMTC) have better processing time efficiency than the (EM). Table 4 shows that HMTC can reduce number of inspection plans down to 88.57% whereas HM can reduce them down to 75.39%.

Table 3 Performance Parameters

| Performance parameters | Range |
|---|---|
| $fn_m$ | 4-7 |
| $V_k$ | 85-180 |
| $Z_k$ | 0.09-0.18 |
| $I_{km}$ | 60-120 |
| $\alpha_k$ | 0.03-0.07 |
| $\beta_k$ | 0.03-0.07 |
| $\Delta_k$ | 0.05-0.09 |
| $x_k$ | 85-120 |

Table 4 shows that the HMTC and HM can produce the optimal solution. However, the cost deviation respectively is only 0.03 and 0.01. On the other hand, the HM still has a bigger savings (75.63%) on processing time than the HMTC. Therefore; the HM can be applied when multistage manufacturing system incorporates more workstations.

Table 4 Performance of the HM in Comparison to EM

| Method | Number of inspection plans considered | Time processing (seconds) | Reducing inspection plans | Saving Time % |
|---|---|---|---|---|
| HM | 252 | 31.2 | 75.39% | 75.63% |
| HMTC | 117 | 36.1 | 88.57% | 71.80% |
| EM | 1024 | 128 | | |

Table 4 Performance of the HM in Comparison to EM (Continued)

| Method | Total cost | | Cost deviation $(\frac{A}{B}-1)100$ | A | |
|---|---|---|---|---|---|
| | | | | HM | HMTC |
| HM | 367126 | | EM | 0.03 | 0.01 |
| HMTC | 362619 | B | HM | | |
| EM | 355589 | | HMTC | | |

## CONCLUSION

1. There is a need for a generalised (composite) model addressing all the similar inspection allocation scenarios. The model can serve as a methaheuristic that will be used to select an appropriate heuristic or algorithm for the solution of a particular AIE problem.
2. The simulation experiment has shown that computational time increases significantly when the number of workstations is more than 20.
3. The majority of the reviewed papers consider analytical models with limited number of work stations between 3 and 5 which is insufficient in practical situations.
4. The inspection allocation problem has been studied comprehensively by using variety of analytical and Monte-Carlo simulation methods rather than combination of both simulation techniques in a simulation-optimisation framework.

## REFERENCES

Bai, D. and H. Yun. 1996. ‘‘Optimal Allocation of Inspection Effort in a Serial Multi-stage Production System’’ *Computers Industrial Engineering*, 30, No.3 (Jul), 387-396.

Ballou, D.P. and H.L. Pazer. 1985. ‘‘Process Improvement Versus Enhanced Inspection in Optimised Systems’’ *International Journal of Production Research*, 23, 1233-1245.

Barad M. 1990. ‘‘A Break Even Quality Level Approach to Location of Inspection Station in a Multi-stage Production Process’’ *International Journal of Production Research*, 28, No.1, 29-45.

Chen, J.D. D. Yao, and S. Zheng. 1998. ‘‘Quality Control for Products Supplied with Warranty’’ *Operations Research*, 46, No.1 (Jan-Feb), 107-115.

Chengalur, I.N., D.P. Ballou and H.L. Pazer. 1992. ‘‘Dynamically Determined Optimal Inspection Strategies for Serial Production Process’’ *International Journal of Production Research*, 30, No.1, 169-187.

Clark, H.J. and J.D.T. Tannock. 1999. ‘‘The Development and Implementation of a Simulation Tool for the Assessment of Quality Economics within a Cell-based Manufacturing Company’’ *International Journal of Production Research*, 37, No.5, 979-995.

Eppen, G.D. and G.E. Hurst. 1974. ‘‘Optimal Location of Inspection Station in a Multi-stage Production Process’’ *Management Science*, 20, No.8 (Apr), 1194-1200.

Estrop, S.J. M.M. Kaye and T.G. Nevell. 1992. ‘‘The use of computer simulation in implementing a manufacturing process quality modelling system’’ *International Journal of Quality and Reliability Management*, 9, No.7, 7-16.

Freiesleben J. 2006. ‘‘Costs and Benefits of Inspection Systems and Optimal Inspection Allocation for Uniform Defect Propensity’’ *International Journal of Quality & Reliability Management*, 23, No. 5 (Oct), 547-563.

Galante, G. and G. Passannanti. 2007. ‘‘Integrated Approach to Part Scheduling and Inspection Policies for a Job Shop Manufacturing System’’ *International Journal of Production Research*, 45, No.22 (Nov), 5177-5198.

Gardner, L.L., M.E. Grant, and L.J. Rolston. 1995. ‘‘Using Simulation to Assess Costs of Quality’’ In *Proceedings of the 1995 Winter Simulation Conference* (Arlington, Virginia). IEEE, Piscataway, NJ., 945-951.

Hadjinicola, C.G. and C.A. Soteriou. 2003. ‘‘Reducing the Cost of Defects in Multi-stage Production Systems: A Budget Allocation Perspective’’ *European Journal of Operational Research*, 145, No.3 (March), 621–634.

Jang, W. and J.G. Shanthikumar. 2002. ‘‘Stochastic Allocation of Inspection Capacity to Competitive Processes’’ *Naval Research Logistics*, 49, No.1 (Apr), 78-94

Jewkes M.E.1995. ‘‘Optimal Inspection Effort and Scheduling for a Manufacturing Process with Repair’’ *European Journal of Operational Research*, 85, No.2 (Sep), 340-351.

Kakade, V. Valenzuela, J. and Smith J. 2004. ‘‘An Optimization Model for Selective Inspection in Serial Manufacturing Systems’’ *International Journal of production Research*, 42, No.18 (Sep), 3891-3909.

Kim, J. and B.S. Gershwin. 2008. ‘‘Analysis of Long Flow Lines with Quality and Operational Failures’’ *IIE Transactions*, 40, 284–296.

Langner, A., D. Montgomery, and W. Carlyle. 2002. ‘‘Solving a Multi-stage Partial Inspection Problem Using Genetic Algorithms’’ *International Journal of Production Research*, 40, No.8, 1923-1940.

Lee, J. and F.F. Chen. 1996. ‘‘Inspection Sequencing and Part Scheduling for Flexible Manufacturing Systems’’ *European Journal of Operational Research*, 95, No.2 (Dec), 344–355.

Lee, J. and S. Unnikrishnan. 1998. ‘‘Planning Quality Inspection Operations in Multi-stage Manufacturing Systems with Inspection Errors’’ *International Journal of Production Research*, 36, No.1 (Jan), 141-155.

Lindsay, G.F. and A.B. Bishop. 1964. ‘‘Allocation of Screening Inspection effort a dynamic programming approach’’10, No.2 (Jan), 342-352.

Mandroli, S., K. Shrivastava, and Y. Ding. 2006. ‘‘A Survey of Inspection Strategy and Sensor Distribution Studies Indiscrete-Part Manufacturing Processes’’ *IIE Transactions*, 38, 309-328.

Narahari, Y. and L.M. Khan. 1996. ‘‘Modelling Re-entrant Manufacturing Systems with Inspection Stations’’ *Journal of Manufacturing Systems*, 15, No.6, 367–378.

Neu, H., T. Hanne, J. Münc, S. Nickel, and A. Wirsen. 2002. ‘‘Creating a Code Inspection Model for Simulation-Based Decision Support’’ *German (SEV Project) and (Pro Sim Project, no.559).*

Penn, M. and T. Raviv. 2007. ‘‘Optimising the Quality Control Station Configuration’’ *Naval Research Logistics*, 54, No.3, 301-314.

Penn, M. and T. Raviv. 2008. ‘‘A polynomial Time Algorithm for Solving a Quality Control Station Configuration Problem’’ *Discrete Applied Mathematics*, 156, No.4 (Feb), 412-419.

Raghavachari, M. and G. Tayi. 1991. ‘‘Inspection Configuration and Reprocessing Decisions in Serial Production Systems’’ *International Journal of Production Research*, 29, No.5 (May), 897-911.

Rau, H. and YH. Chu. 2005. ‘‘Inspection Allocation Planning with Two Types of Workstation WVD and WAD’’ *International Journal of Advanced Manufacturing Technology*, 25, No.9-10 (May), 947-953.

Rau, H., YH. Chu, and KH. Cho. 2005. ‘‘Layer Modelling for the Inspection Allocation Problem in Re-entrant Production Systems’’ *International Journal of Production Research*, 43, No.17 (Sep), 3633-3655.

Raz, T., Y.T. Herer, and A. Grosfeld-Nir. 2000. ‘‘Economic Optimization of Off-Line Inspection’’ *IIE Transactions*, 32, 205–217.

Raz, T. and M. Kaspi. 1991. ‘‘Location and Sequencing of Imperfect Inspection in Serial Multi-stage Production Systems’’ *International Journal of Production Research*, 29, No.8, 1645-1659.

Rebello, R., A. Agnetis, and P.B. Mirchandani. 1995. ‘‘Specialized Inspection Problems in Serial Production Systems’’ *European Journal of Operational Research*, 80, No.2 (Jan), 277-296.

Saxena, S. C.M., Chang, H.B., Chow and J. Lee. *1990.* ‘‘Evaluation of Heuristics for Inspection Station Allocation in Serial Production Systems’’ In Proceedings Winter Simulation Conference, (New Orleans, Louisiana 9-12 Dec). IEEE Piscataway, NJ., 919-922.

Siemiatkowski, M., and W. Przybylski. 2006. ‘‘Simulation Studies of Process Flow with in-Line Part Inspection in Machining Cells’’ *Journal of Materials Processing Technology*, 171, No. 1 (Jan), 27–34.

Shiau YR. 2002. ‘‘Inspection Resource Assignment in a Multi-stage Manufacturing System with an Inspection Error Model’’ *International Journal of Production Research*, 40, No.8 (May), 1787-1806.

Shiau, YR. 2003a. ‘‘Inspection Allocation Planning for a Multiple Quality Characteristics Advanced Manufacturing Technology’’ *International Journal of Advanced Manufacturing Technology*, 21, No.7 (May), 494-500.

Shiau, YR. 2003b. ‘‘Quick Decision-Making Support for Inspection Allocation Planning with Rapidly Changing Customer Requirements’’ *International Journal of Advanced Manufacturing Technology*, 22, No.9-10(Nov), 633-640.

Shiau, YR., MH. Lin, and WC. Chung. 2007. ‘‘Concurrent Process Inspection Planning for a Customized Manufacturing System Based on Genetic Algorithm’’ *International Journal Advanced manufacturing Technology*, 33, No. 7-8 (Jul), 746-755.

Taneja, M. and N. Viswanadham. ‘‘Inspection Allocation in Manufacturing Systems a Genetic Algorithm Approach’’ In *Proceedings of the 1994* IEEE California International Conference on Robotics and Automation, (San Diego, California, 8-13 May). 3537-3542.

Tannock, J.D.T. 1995. ‘‘Choice of Inspection Strategy Using Quality Simulation’’ *International Journal of Quality & Reliability Management*, 12, No.5, 75-84.

Tannock, J.D.T. 1997. ‘‘An Economic Comparison of Inspection and Control Charting Using Simulation’’ International Journal of Quality & Reliability Management, 14, No.7, 687-699.

Tannock, J. and S. Saelem. 2007. ‘‘Manufacturing Disruption Costs Due to Quality Loss’’ *International Journal of Quality & Reliability Management*, 24, No.3, 263-278.

Tayi, G.K. and D.P. Ballou. 1988. ‘‘An Integrated Production Inventory Model with Reprocessing and Inspection’’ *International Journal of production Research,* 26, No.8 (Aug), 1299-1315.

Valenzuela, F.J., S.J. Smith, and S.J. Evans. 2004. ‘‘Allocating Solder-Paste Printing Inspection in High Volume Electronics Manufacturing’’ *IIE Transactions*, 36, 1171–181.

Van Volsem, S., W. Dullaert and H. Van landeghem. 2007. ‘‘An Evolutionary Algorithm and Discrete Event Simulation for Optimising Inspection Strategies for Multi-stage Processes’’ *European Journal of Operational Research*, 179, No.3 (June), 621-633.

Yao, D.D. and S. Zheng. 1999. ‘‘Sequential Inspection under Capacity Constraints’’ *Operations Research*, 47, No.3 (May-June), 410-422.

**BIOGRAPHY**

**ALI SHETWAN** received the B.Sc. from Alfateh University, Tripoli, Libya in 1986 in Industrial Engineering and M.Sc. degrees from the Higher Institute of Industry Misurata, Libya in Production and Quality Engineering in 1999. Since 2008 he is PhD student in school of engineering at the group of Prof. Vitanov, at Durham University. He is mainly interested in quality and production.

**Valentin VITANOV** is a Professor of Design Manufacture and Management in the School of Engineering and Computer Sciences at the University of Durham UK. He received his PhD in Industrial Automation and Robotics from the St. Petersburg Electrotechnical University (Russia). He is a Fellow of the Institute of Engineering and Technology, UK. His research interests include applied statistics and robust design, manufacturing systems simulation and optimisation, operational research and autonomous robotics.

# IMPROVING THE JOB-SHOP WORKLOAD CONTROL THROUGH ORDER ACCEPTANCE AND DUE-DATE NEGOTIATION

Maria do Rosário Alves Moreira
Faculdade de Economia, Universidade do Porto
R. Dr. Roberto Frias, 4200-464 Porto, Portugal
E-mail: mrosario@fep.up.pt

**KEYWORDS**
Job-shop; acceptance decision; workload control

## ABSTRACT

Work flows in a job-shop are determined not only by the release load but also by the number of accepted orders. In this paper the common assumption of accepting all incoming orders regardless of shop condition is relaxed. Instead of placing the orders in a ‘pre-shop pool’ queue, as in previous research, orders that arrive at the shop, when it is highly congested, may be immediately rejected or their due dates may be negotiated. This paper explores the idea of controlling the workload since the acceptance/rejection stage. A new acceptance/rejection rule is proposed, and tests are conducted to study the sensitivity of job-shop performance to different order acceptance parameters, like the tolerance of the workload limit and the due date extension acceptance. The effect of the negotiation phase on the job-shop performance is evaluated using a simulation model of a generic random job-shop that allow us to conclude that having a negotiation phase prior to rejection improves almost all workload performance measures. Different tolerances of the workload limit slightly affect the performance of the job-shop.

## INTRODUCTION

Workload and input-output control have been attracting increased attention among researchers in recent years. However, these studies focus mostly on workload control only after the orders have been accepted, i.e. traditionally the workload in a job-shop is controlled at the order release stage. Alternatively, decisions on workload control can be made earlier, at the stage of order acceptance or rejection. It may be seen as a rather extreme form of workload control, but if this decision is made using an appropriate rejection rule, it may be advantageous for the system as a whole. When capacity is fixed and demand is high the company has to decide which orders to accept and which orders to reject. Rejecting an order may be more favourable to the goodwill of the company than accepting all orders regardless of capacity restrictions and, consequently, completing a significant percentage after their due date.

A new acceptance/rejection rule is proposed, and tests are conducted to study the importance of having a negotiation stage before definitely rejecting the order. The sensitivity of the shop performance to different order acceptance parameters is also analysed. The proposed acceptance-rejection rule, called DDN (Due Date Negotiation), takes into account three types of information: (i) the total workload of jobs in-process plus jobs waiting in the pre-shop pool; (ii) a pre-defined tolerance of the workload limit (called negotiation margin); and (iii) the order’s due date. The idea behind this new rule is to allow for the acceptance of new orders when the workload limit is exceeded by only a small percentage. With this rule, the number of rejected orders decreases and the shop-floor congestion is controlled through the due date negotiation.

The operational performance measures used were related to delivery and workload performance. Since performance measures are influenced not only by the parameters of the acceptance rule but also by other decisions, a benchmark rule and a *good* rule previously presented in the literature are considered for each of these decisions. The simulation experiments were performed to investigate whether the negotiation phase improves the job-shop workload control by comparing selected performance measures in the two situations (with and without negotiation). The results allow us to conclude that having a negotiation phase prior to rejection improves almost all workload and delivery related performance measures. The simulation results also show that the shop performance is slightly sensitive to the customer acceptance probability and that different tolerances of the workload limit affect little the performance of the job-shop.

This paper has three main objectives: firstly, to present, simulate and test a new decision rule (to accept or not an incoming order), secondly, to investigate whether the order negotiation phase improves the shop-floor performance, and thirdly, to study the sensitivity of the shop performance to different order acceptance parameters, like the tolerance of the workload limit and the due date extension acceptance.

The remainder of this paper is structured as follows. In the following section related research on order acceptance is reviewed. Then, the proposed acceptance/rejection rule is presented, together with a description of the simulation manufacturing environment in which it is tested. The research methodology is outlined afterwards. Following this, the results of the simulation experiments are presented and discussed, and in the final section some conclusions and possible directions for future research are highlighted.

## RELATED WORK

Despite a clear early concern about workload control (Wight, 1970), order acceptance received limited attention in the literature until the last decade. Order acceptance deals with the decision to either accept or reject a customer’s order based on the availability of sufficient capacity to complete it as close as possible to its due date. Most papers have focused on alternative methods for releasing jobs to the shop-floor as the first mechanism to control workload. A good survey and classification of the research in the field of order review and release (ORR) can be found in Bergamaschi et al. (1997).

Input control considering the rejection of a portion of demand was first presented in queuing theoretic models (Scott, 1969, 1970; Miller, 1969, e.g.) that consider an *n*-server queuing system with *m* customer classes. This form of input control is common and occurs quite often in service systems: whenever the number of customers in the queue reaches its maximum capacity (e.g. due to a limited waiting room) no more customers are admitted into the system. In manufacturing systems, order acceptance decisions are often based on the workload content of the order, related to the available workload. On the experimental side, Philipoom and Fry (1992), Ten Kate (1994), Raaymakers et al. (2000a, 2000b), Ivanescu et al. (2002) and Ebben et al. (2005) compare different order acceptance strategies (algorithms) using simulation. Other approaches to this problem include mathematical programming and the use of meta-heuristics to examine order acceptance decisions when capacity is limited. Slotnik and Morton (1996), develop an optimal branch-and-bound procedure in order to maximize revenues. Lewis and Slotnik (2002) and Slotnik and Morton (2007) extend the one-period, deterministic model to a multi-period problem.

We study a job shop problem rather than the single-machine problem in the existing literature, e.g. Ten Kate (1994), Slotnik and Morton (1996, 2007), Lewis and Slotnik (2002). Contrary to the existing literature where all jobs are available initially (Slotnik and Morton, 1996, 2007 or Alidaee et al., 2001) we insert new jobs into an existing schedule without altering previously promised due dates. While a considerable amount of research has been done on designing order acceptance policies or algorithms to optimize some performance measure, our focus is on studying the impact of a negotiation phase in the order acceptance process.

## THE ORDER ACCEPTANCE/REJECTION RULE

In this section the acceptance/rejection decision is placed in the global decision making process and the proposed acceptance rule is described in detail. The production control system, for the kind of job-shop considered, consists of four stages: 1) acceptance, negotiation or rejection of an order, 2) due date assignment, 3) order release, and 4) order dispatch. The accept/negotiate/reject decision is made when a customer places an order. In this paper two rules are considered: total acceptance (TA), used as a benchmark, and the proposed rule, the due date negotiation (DDN). The decision about the due date assignment is made simultaneously with the acceptance decision, and a negotiation with the customer may occur. We will consider only one due date assignment rule because, by varying the planning parameter, it is possible to convert one rule into another. The total work content (TWK) rule defines the due date by adding a certain amount, representative of the time that the job will need to be completed, to the order's arrival date:

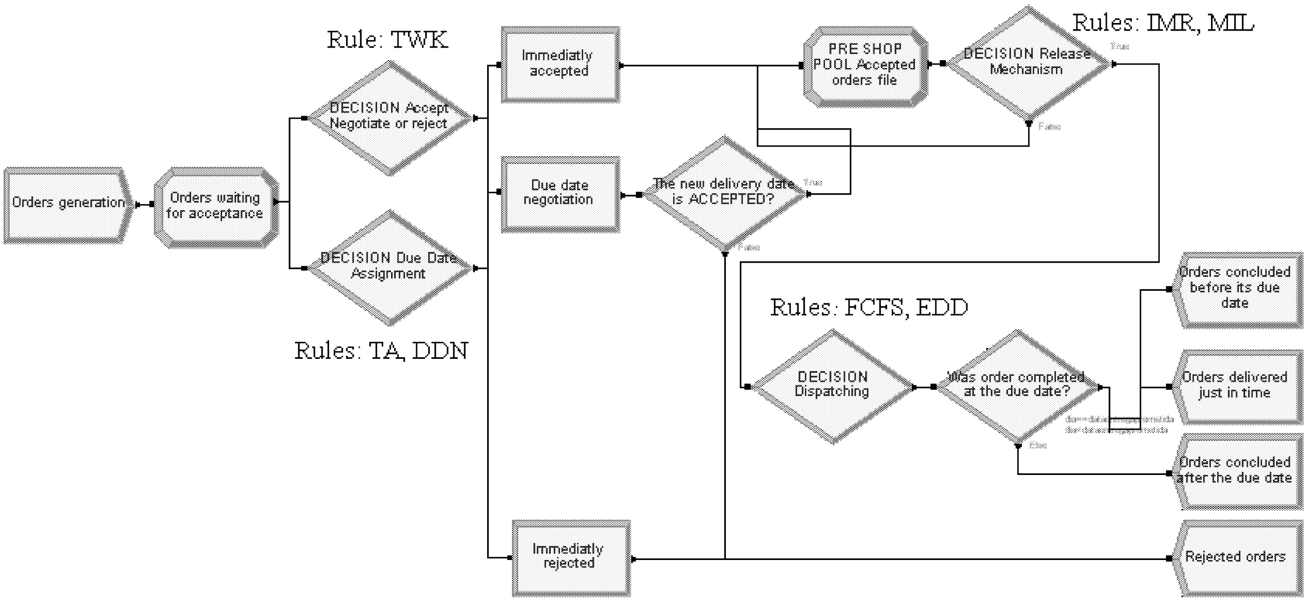
$$DD_i = AD_i + k_{TWK} \times P_i \ , \tag{1}$$

where: $DD_i$: due date of job i;
$AD_i$: job i arrival date;
$P_i$: processing time of job i;
$k_{TWK}$: planning factor.

After an order has been accepted, it is placed in a pre-shop pool file. The order release rule defines when a release must take place and which of the orders will be released to the shop-floor. Two order release rules are considered: immediate release (IMR) and modified infinite loading (MIL). The IMR release rule is used as a benchmark: as soon as an order is accepted it is released to the shop-floor. The MIL rule was proposed by Ragatz and Mabert (1988) as an extension of the backward infinite loading rule (BIL), which consists in deducting from the due date the expected job flow time. It is similar to the BIL rule (because it ignores the shop capacity), but it has more information to predict the job flow time since it includes a factor about the present work on the shop. MIL determines the job release date as follows:

$$RD_i = DD_i - k_{1MIL} \times n_i - k_{2MIL} \times Q_i \ , \tag{2}$$

where: $RD_i$: release date of job i; $DD_i$: due date of job i;
$n_i$: number of operations of job i;
$Q_i$: number of jobs in queue on job i routing;
$k_{1MIL}$, $k_{2MIL}$: planning factors.

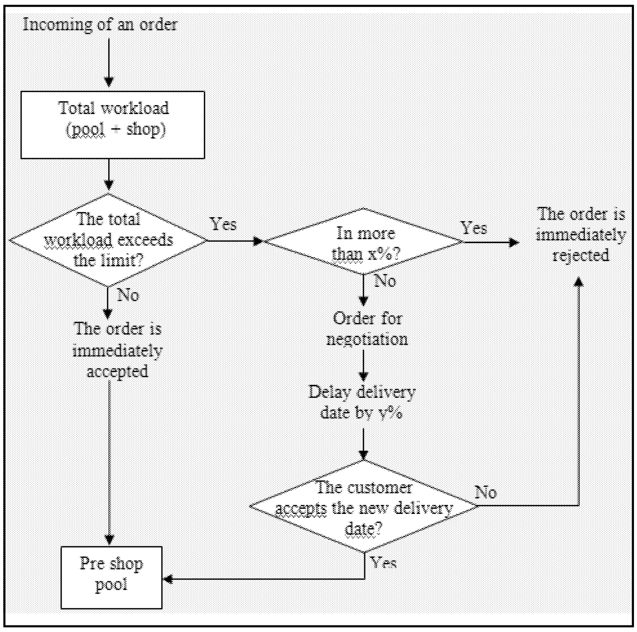


Rule: TWK
Rules: TA, DDN
Rules: IMR, MIL
Rules: FCFS, EDD
Orders generation
Orders waiting for acceptance
DECISION Accept Negotiate or reject
DECISION Due Date Assignment
Immediatly accepted
Due date negotiation
Immediatly rejected
The new delivery date is ACCEPTED?
True
False
PRE SHOP POOL Accepted orders file
DECISION Release Mechanism
True
False
DECISION Dispatching
Was order completed at the due date?
Else
Orders concluded before its due date
Orders delivered just in time
Orders concluded after the due date
Rejected orders

Figure 1: Multiple Decision-making Scheme in Arena

Once a job is released to the shop-floor, its progress is controlled by the selected dispatching rule. We will consider the first-come-first-serve rule (as a benchmark) and the earliest due date (EDD) rule. When all processing has been completed, the order is placed in a finished-goods inventory until its delivery (due) date. Figure 1 shows these four decisions and the relationships among them, using the Arena software layout, the software package used in the simulation experiments.

The proposed due date negotiation (DDN) rule works as follows: when an order arrives, the total workload in the shop (considering the jobs on the shop-floor and the jobs waiting for release) is computed. If it is lower than a pre-defined limit, the order is immediately accepted. However, if the total workload exceeds that limit, one of two decisions may be made: the negotiation of the due date or the rejection of the order. Due date negotiation occurs whenever the pre-defined limit is exceeded by only a certain (small) percentage (the negotiation margin). In this case, an extension of the order's delivery date is proposed to the customer; if the customer accepts the new delivery date(which happens with a certain probability), the order is accepted. If there is no negotiation or if the customer does not accept the new delivery date the order is rejected. In Figure 2 we can see how the DDN rule works.



Incoming of an order
Total workload (pool + shop)
The total workload exceeds the limit?
Yes
No
The order is immediately accepted
In more than x%?
Yes
No
The order is immediately rejected
Order for negotiation
Delay delivery date by y%
The customer accepts the new delivery date?
No
Yes
Pre shop pool

Figure 2: Due Date Negotiation (DDN) Rule

The DDN rule has four parameters that must be carefully defined. We will test the sensitivity of the shop performance to some of them (tolerance of the workload limit - $L_{Max}$, negotiation margin - *Nm*, and the due date extension acceptance - *Pa*). The other (delay of the original due date - *Ddd*) remain unchanged. Table 1 summarizes the parameters that need to be specified.

Table 1: DDN parameters

| | |
|---|---|
| $L_{Max}$ | Workload limit to accept an order |
| *Ddd* | Delay (in percentage) of the original delivery date |
| *Nm* | Negotiation margin |
| *Pa* | Percentage of customers that accepts the new DD |

The workload limit to accept an order ($L_{Max}$) depends on the order's delivery date and on the number of machines the order has to visit in its routing. Equation (3) shows how it is computed.

$$L_{Max} = \frac{89*(\text{due date} - \text{present date})}{\text{Number of machines that are in the order's routing}} \quad (3)$$

On the one hand, the greater the difference between the due and the present date the greater the chance to deliver the order in time. On the other hand, the greater the number of machines the order has to visit the more the shop-floor will be work-loaded. If the machines have a high utilization, orders that have a less complex routing will have priority. $L_{Max}$ is defined so that the mean percentage of rejected orders is 5%, when the DDN, the IMR and the FCFS rules are in use. The delay of the original delivery (due) date is defined as a percentage of the original due dates, not in days, and is equal to 5%. This percentage should not be very high to be a good representation of what happens in real manufacturing systems. The negotiation margin is the amount (in percentage) that the workload limit may be exceeded without an order being rejected. As we want to control the workload, this percentage should be very small. As a benchmark to compare the situation with and without negotiation, we will set *Nm* = 10%. Later on, we will test the sensitivity of job-shop performance to different tolerances of the workload limit. The parameter *Pa* corresponds to the percentage of customers that accept the extension of the original due date. *Pa* is set at 70%, and the sensitivity of job-shop performance to different values of *Pa* is also tested.

## RESEARCH METHODOLOGY

### Shop-floor characteristics

Orders are assumed to arrive according to a Poisson process. Besides the widespread use of the Poisson distribution, there is some theoretical evidence that it provides a good approximation for the arrival process (Albin, 1982). The routing for each order and the processing time at each station is generated at this stage. The routing is purely random: the number of operations follows a discrete uniform probability distribution between one and six machines. The order has an equal probability of having its first operation in any of the six machines and of going to the other machines, until being completed. After the definition of the job characteristics, the order is placed in a pending (for acceptance) orders file.

The characteristics of the hypothetical job-shop are identical to those used by Melnyk and Ragatz (1989): the shop consists of six work centres, operating 40 hours per week; each work centre contains a single machine that can process only one job at a time, and no preemptions are allowed; job routings are random, with no return visits. Order arrivals follow a Poisson process with a mean of 1 order per hour. The processing time distribution for all six machines is identical: exponential with a mean of 1.5 hours. These characteristics result in a steady state utilization rate of 87.5% for each work centre and for the shop as a whole.

### Simulation model and experimental factors

Simulation was used to study the proposed rule and the impact of a negotiation phase. The applicability of simulation in the general area of manufacturing systems analysis is well known since it can handle complex stochastic systems in arbitrary detail (Grant, 1988). The simulation model was developed using the software Arena 7.1 (Kelton et al., 2004).

In testing the acceptance/rejection rule, it is important to assess whether the performance is affected by other factors in the planning system, such as the order release and the dispatching rules being used. Therefore, a full $2 \times 2 \times 2$ experimental design was used: the two accept/reject rules described above are simulated in combination with the two order release rules and the two priority dispatching rules presented. The value of the planning factor ($k_{TWK}$) in the due date formula (1) is set at 38, because, with this value, the percentage of tardy jobs is about 10%, when the DDN, IMR, and FCFS rules are simulated.

In testing the sensitivity of job-shop performance to different order acceptance parameters, a full $1 \times 2 \times 2 \times 8$ experimental design is used: the DDN accept/reject rule is simulated in combination with two order release rules, two priority dispatching rules and eight levels for the negotiation margin (*Nm* = 20%, 15%, 12.5%, 10%, 7.5%, 5%, 2.5% and 0%). *Nm* = 0% corresponds to the situation where negotiation does not occur, but the rejection can take place if the workload limit is surpassed. In the other extreme case, the workload limit can be exceeded by 20% without having an immediate rejection. The sensitivity of job-shop performance to the percentage of customers that accept the new delivery date is also tested. Here, we use a 24 experimental design: the DDN accept/reject rule is simulated in combination with two order release rules, the two priority dispatching rules and six levels for the due date extension acceptance (the percentage of customers that accept the new delivery date varies between 50% and 100%). When *Pa* is 100%, all customers accept the due date extension. Similarly, when *Pa*=50% only half of the customers accept the new delivery date.

### Performance measures

In order to assess the impact of the decision rules on manufacturing performance, specific performance criteria must be selected. Five measures of job-shop performance are considered. These measures are broken down in two categories:

(i) Due date related performance measures, which are indicative of customer satisfaction and deliverability: mean tardiness and percent tardy.

(ii) Workload related performance measures, which are used to evaluate the impact of the load observed on the shop-floor: mean wait time in final products inventory, mean queue time in the shop-floor and machine utilization.

### Data collection

During simulation runs data are collected with reference to the steady state of the system. In order to remove the effects of the warm-up period, several runs of the simulation model were made to see when the steady state was reached. Performance criteria and utilization levels reached steady state after approximately 4,000 (simulated) working hours. However, all statistics were set to zero and restarted after a warm-up period of 10,000 simulated hours. Statistics were, then, collected for 90,000 hours. Ten replications were performed for each set of experimental conditions.

## COMPUTATIONAL RESULTS

In this section, we present the main results of the experiments. The analysis is divided in three parts: the first one discusses if the negotiation phase improves the workload control; the second one presents the main results of the sensitivity analysis of the *Nm* parameter; in the third one the results of the sensitivity analysis to the *Pa* are presented.

### Order negotiation phase

To find out if order negotiation improves the shop-floor workload control we compare the results on the selected performance measures of the TA rule (inexistence of order rejection) with the DDN one. Figure 3 and Figure 4 show the due date and workload related performance measures, respectively.

The simulation was made for the eight possible combinations to assess if the differences observed are due to the existence of the order negotiation or are due to other factors. For each rule (TA and DDN) we first compare the results obtained for the mean tardiness and percentage of tardy jobs in each experimental design. We can observe that the DDN rule results in a better delivery performance in all of the possible combination of decision rules in both performance measures analysed. The improvement is larger when the MIL order release rule is employed. When we look at the performance measures related with the workload (Figure 4), we notice that the order negotiation, in three of the combinations, allows the order to spend less time in queues inside the shop-floor. And when the order is released as soon as it enters the job-shop and the dispatching rule is EDD, the mean queue time in the shop-floor is almost the same. The only workload measure that has worst results when the DDN is used is the mean wait time in final products inventory (the mean earliness). Another advantage of the order negotiation is that it implies a slight decrease in the percentage of machine utilization (see second graph of Figure 4). It is known that one of the constraints job-shops have is the lack of capacity, due to the unstable routings and demand of their products. A decrease in the utilization is good for the shop-floor, because it becomes easier to control the workload. It is important to notice that in every performance measure (except "mean earliness"), when the MIL rule is employed, independently of the dispatching rule in use, the values for the DDN rule are consistently better. Comparing the mean values for the two acceptance rules the hypothesis that DDN performs better cannot be rejected (with a significance level of 5%).

### Sensitivity analysis of *Nm* and of *Pa*

To analyse the sensitivity of the selected performance measures to different values of the negotiation margin, all the other parameters are kept fixed. As mentioned earlier, *Nm* varies from 20% to 0%. Analysing the different combination of decision rules, it can be seen that the only performance measure that is sensitive to *Nm* is the mean tardiness in DDN-MIL-FCFS. We also see slight sensitivity in mean wait time in final products inventory in the design DDN-IMR-EDD. The percentage of customers that accept the new delivery date varies from 50% to 100%. Order negotiation allows for a significant improvement in workload and delivery measures, but performance is not very sensitive to the variation of the parameters, namely the negotiation margin and the due date extension acceptance.

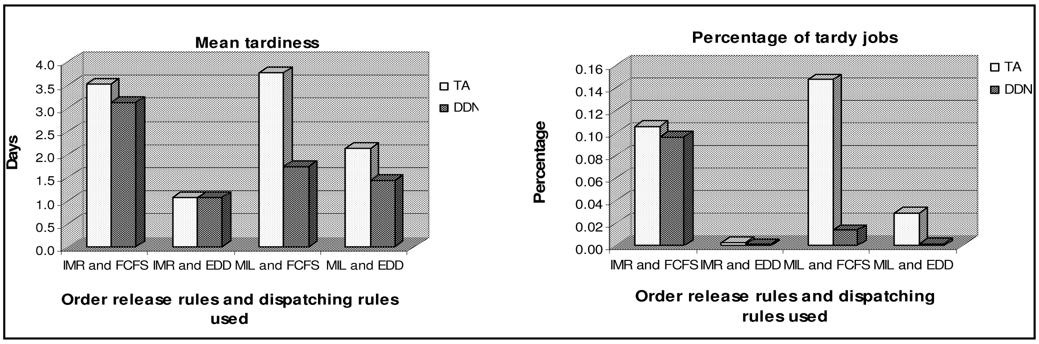


Figure 3: Due Date Related Performance Measures

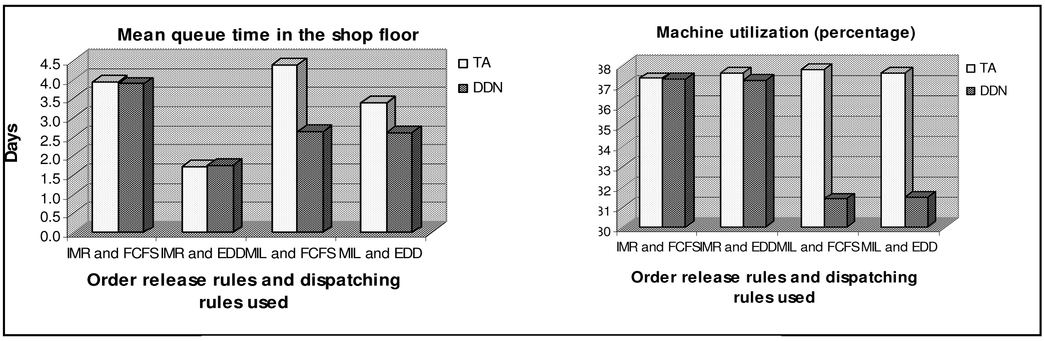


Figure 4: Workload related performance measures

## SUMMARY AND CONCLUSIONS

This paper explores the idea of controlling the workload since the stage of accepting or rejecting incoming orders. A new acceptance/rejection rule, DDN or due date negotiation, is proposed. It evaluates the impact of different tolerances of the workload limit (*Nm*, the negotiation margin) on the shop performance using a simulation model of a generic random job-shop and a full factorial experimental design. It also tests the influence of various probabilities of acceptance (*Pa*) by the customers of the new delivery date on the shop performance.

In testing the acceptance/rejection rule and its parameters it is important to assess whether the shop performance is affected by other factors in the planning system, such as the order release and the dispatching rule being used. Therefore, a full experimental design is used: the acceptance/rejection rule described above is simulated in combination with two order release rules (the *immediate release* — a benchmark rule — and the *modified infinite loading* rule) and two priority dispatching rules (the *first come first served* — a benchmark rule — and the *earliest due date* rule). The extensive simulation experiments allow us to conclude that both the workload and the delivery performance measures improve with the use of a rule that includes a negotiation phase. We also see that different tolerances of the workload limit affect, to some extent, the performance of the job-shop.

## REFERENCES

Albin, S.L., 1982. On Poisson approximations for superposition arrival processes in queues. *Management Science*, 28 (2), 126-137.

Alidaee, B., Kochenberger, G.A. and Amini, M.M., 2001. Greedy solutions of selection and ordering problems. *European Journal of Operational Research*, 134 (1), 203-215.

Bergamaschi, D., Cigolini, R., Perona, M. and Portioli A., 1997. Order review and release strategies in a job-shop environment: a review and classification. *Int. Journal of Production Research*, 35 (2), 399-420.

Ebben, M.J., Hans, E.W. and Olde-Weghuis F.M., 2005. Workload based order acceptance in job-shop environments. *OR Spectrum*, 27, 107-122.

Grant, F.H., 1988. Simulation in designing and scheduling manufacturing systems. In Design and analysis of integrated manufacturing systems, National Academy Press: Washington DC

Ivanescu, C.V., Fransoo, J.C. and Bertrand J.W., 2002. Makespan estimation and order acceptance in batch process industries when processing times are uncertain. *OR Spectrum*, 24 (4), 467-495.

Kelton, W.D., Sadowski, R.P. and Sturrock, D.T., 2004. Simulation with Arena, McGraw-Hill: New York.

Lewis, H.F. and Slotnik, S.A., 2002. Multi-period job selection: planning work loads to maximize profit. *Computers & Operations Research*, 29 (8), 1081-1098.

Melnyk, S.A. and Ragatz, G.L., 1989. Order review/release: research issues and perspectives. *Int. Journal of Production Research*, 27 (7), 1081-1096.

Miller, B.L., 1969. A queueing reward system with several customer classes. *Management Science*, 16 (3), 234-245.

Philipoom, P.R. and Fry, T.D., 1992. Capacity-based order review/release strategies to improve manufacturing performance. *Int. Journal of Production Research*, 30 (11), 2559-2572.

Raaymakers, W.M., Bertrand, J.W. and Fransoo J.F., 2000a. The performance of workload rules for order acceptance in batch chemical manufacturing. *Journal of Intelligent Manufacturing*, 11 (2), 217-228.

Raaymakers, W.M., Bertrand, J.W. and Fransoo J.F., 2000b. Using aggregate estimation models for order acceptance in a decentralized production control structure for batch chemical manufacturing. *IIE Transactions*, 32 (10), 989-998.

Ragatz, G.L. and Mabert, V.A., 1988. An evaluation of order release mechanisms in a job-shop environment. *Decision Sciences*, 19, 167-189.

Scott, M. A., 1969. A queueing process with some discrimination. *Management Science*, 16 (3), 227-233.

Scott, M., 1970. Queueing with control on the arrival of certain type of customers. *CORS Journal*, 8 (2), 75-86.

Slotnik, S.A. and Morton, T.E., 1996. Selecting jobs for a heavily loaded shop with lateness penalties. *Computers & Operations Research*, 23 (2), 131-140.

Slotnik, S.A. and Morton, T.E., 2007. Order acceptance with weighted tardiness. *Computers & Operations Research*, 34 (10), 3029-3042.

Ten Kate, H.A., 1994. Towards a better understanding of order acceptance. *Int. Journal Production Economics*, 37 (1), 139-152.

Wight, O., 1970. Input/Output control: a real handle on lead time. *Production and Inventory Management Journal*, 11 (3), 9-30.

Wu, M.C. and Chen, S.Y., 1996. A cost model for justifying the acceptance of rush orders. *Int. Journal Production Research*, 34 (7), 1963-1974.

### AUTHOR BIOGRAPHY

**M. ROSÁRIO ALVES MOREIRA** was born Portugal and went to the University of Porto, where she studied Management at the Faculty of Economics and obtained her degree in 1994. In 1997, she obtained her master degree in Economics at the University of Porto. After doing the Management Quantitative Methods Master Course, she started her PhD research in the field of workload and input-output control, obtaining her PhD degree in 2005. She is teaching Operation Research and Operations Management at Faculty of Economics since 1996 until the present date.

# OPTIMIZATION OF CAR REPAIR PROCESSES BY SCATTERED CONTEXT GRAMMARS APPLICATION

Šárka Květoňová
Dušan Kolář
Faculty of Information Technology
Brno University of Technology
Božetěchova 2
612 66 Brno, Czech Republic
E-mail: {kvetona, kolar}@fit.vutbr.cz

## KEYWORDS

Processes and Sub-processes, Scattered Context Grammars, Haskell, Optimization, Scheduling, Chomsky Hierarchy.

## ABSTRACT

The aim of this paper is to present an application of grammars/languages as a suitable way (tool) for description of complex processes (process management domain).

It describes how context grammars, in particular, *scattered context grammars* can be used to model and, subsequently, to control complex process system and its optimization, e.g. car repair service. Our intention is to clarify the whole process (include many others sub-processes and individual activities with different dependencies and relations) and controlling through the models defined by a scattered context grammar. Practical realization (implementation) is given in Haskell (functional programming language with reference transparency).

The created car repair service model is possible, in future, to extend with modelling of real resources.

## INTRODUCTION

It is quite well accepted that the need of efficient process management is required together with increasing complexity of resource consumption. Nowadays, processes are going to be more and more complicated, there are many sub-processes and individual activities demanding many resources with different abilities, alignments, capacities, etc. Processes can be dependent and/or independent on the others; they can be realized concurrently and/or sequentially. Thus, it is necessary to find a new suitable and a more efficient tools, techniques, methods, and possibilities for optimal process management.

The paper presents one of the possibilities how we can describe a sample complex process. In our case, it is a car repair service process. We analyze four basic blocks of processes and four workrooms in a car maintenance service.

In the following section, the main foundations will be given that are crucial for expression and comprehension of a mutual context among terms of formal languages and process management.

## TECHNICAL BACKGROUND

First of all, some terms relating to process management and theoretical aspects of formal languages and grammars are presented. Nevertheless, it is expected that a reader is familiar with the concept. Deeper explanation of the terms can be found in (Češka & Rábová 1988, Meduna 2000, Meduna, & Techet, 2009, Meduna & Švec 2005, Greibach & Hopcroft 1969, Fernau 1996).

### Process management

*Definition 1.* Process management is the ensemble of activities of planning and monitoring the performance of a process. Especially in the sense of business process, often confused with reengineering (Becker 2003).

*Definition 2.* A process is a series of actions bringing about a result. It is a complex of mutually connected resources and activities, which change inputs to outputs. At present, activities and resources under the project are managed almost entirely like processes, see (Anbari 2005). From other point of view, a process is a specific ordering of work activities across time and place with a beginning, an end, and clearly defined inputs and outputs: a structure for action. See (Sparx Systems 2004).

*Definition 3.* A business process is a collection of related, structured activities that produce a service or product that meets the needs of a client. These processes are critical to any organization as they generate revenue and often represent a significant proportion of costs.

*Definition 4.* Business Process Management (BPM) is a management approach focused on aligning all aspects of an organization with the wants and needs of clients. It is a holistic management approach that promotes business effectiveness and efficiency while striving for innovation, flexibility, and integration with technology (Smart, Maddern, & Maull 2001).

*Definition 5.* A workflow is a model to represent real work for further assessment, e.g., for describing a reliably repeatable sequence of operations. More abstractly, a workflow is a pattern of activity enabled by a systematic organization of resources, defined roles and mass, energy and information flows, into a *work process* that can be documented and learned. Workflow consists of a sequence of connected steps. It is a depiction of a sequence of operations, declared as work of a person, a group of persons, an organization of staff, or one or more simple or complex mechanisms. For control purposes, workflow may be a view on real work under a chosen aspect, thus serving as a virtual representation of actual work. The flow being described often refers to a document that is being transferred from one step to another (Fischer 2007).

*Definition 6.* Process analysis is an approach that helps managers improve the performance of their business activities. It can be a milestone in continuous improvement (Trischler 1996)

**Grammars**

*Definition 7.* A grammar, G, is a quadruple G = (N, T, P, S), where N is a final set of non-terminals, T is a final set of terminals, $T \cap N = \varnothing$, P is a final set of production rules, it is a subset of $(N \cup T)^* N (N \cup T)^* \times (N \cup T)^*$, an element $(\alpha, \beta) \in P$ will be written $\alpha \rightarrow \beta$, and the symbol S is the starting non-terminal, $S \in N$.

To define language defined by a grammar, we have to define the term derivation. A definition of it, together with a definition of language defined by a grammar, follows:

*Definition 8.* If $\alpha \rightarrow \beta \in P$ and $u, v, \beta \in (N \cup T)^*$, $\alpha \in (N \cup T)^* N (N \cup T)^*$, then $u \alpha v \Rightarrow u \beta v [\alpha \rightarrow \beta]$ or, simply, $u \alpha v \Rightarrow u \beta v$ is called a simple derivation. In the standard manner, extend $\Rightarrow$ to $\Rightarrow^n$, where $n \geq 0$; then, based on $\Rightarrow^n$, define $\Rightarrow^+$ and $\Rightarrow^*$, a (general) derivation.
The language of G, L(G), is defined as $L(G) = \{w \in T^* \mid S \Rightarrow^* w\}$.

By restricting the form of production rules used for a grammar definition, we can recognise several kinds of grammars and languages defined by such grammars. Next are defined some kinds of them.

*Definition 9.* A context-sensitive grammar, G, restricts P (a finite set of productions) such a way, so that for every $\alpha \rightarrow \beta \in P$: $| \alpha | \leq | \beta |$. If an empty string ($\varepsilon$) is in the language a special rule is allowed in P: $S \rightarrow \varepsilon$, where S is the starting non-terminal. A language, L, is context-sensitive if and only if L = L(G), where G is a context-sensitive grammar.

*Definition 10.* A context-free grammar, G, restricts P (a finite set of productions) to the form $A \rightarrow x$, where $A \in N$ and $x \in (N \cup T)^*$. If there are several rules of the form $A \rightarrow \alpha_1$, $A \rightarrow \alpha_2$, . . . , $A \rightarrow \alpha_n$, where $A \in N$, $\alpha_i \in (N \cup T)^*$ for $i \in \{1, . . . , n\}$, $\alpha_i$ are mutually different, then we can write them in the form $A \rightarrow \alpha_1 | \alpha_2 | . . . | \alpha_n$.

A language, L, is context-free if and only if L = L(G), where G is a context-free grammar.

An example of a simple context-free grammar describing processing of a simplified product order is presented next:

Let G = (N, T, P, S) be a grammar defining such an order processing, then

N = {S, A, B, C, D, E, F}
T = {<order evaluation>, <availability on stock>, <raw material purchase>, <production preparation>, <production>, <delivery>, <invoicing>, <payment verification>}
P = {S → <order evaluation> A, A → ε, A → <availability on stock> B, B → D, B → <raw material purchase> C, B → <production preparation> C, C → <production> D, D → <delivery> <invoicing> E, E → <payment verification> F, F → ε, F → E}

For better readability, we usually join right-hand-sides of the rules from P together and delimit them by a pipe, |. Moreover, the content of the set P can also be written in a more readable form, where we devote each line to one non-terminal on the left-hand-side of the production rule:

```
P = {
        S → <order evaluation> A,
        A → ε
          |  <availability on stock> B
        B → D
          |  <raw material purchase> C
          |  <production preparation> C
        C → <production> D
        D → <delivery> <invoicing> E
        E → <payment verification> F
        F → ε
          |  E
}
```

Definitions of other grammar categories continue next.

*Definition 11.* A regular grammar, G, restricts P (a finite set of productions) to the form $A \rightarrow x$, where $A \in N$ and $x \in T(N \cup \{\varepsilon\})$.
A language, L, is regular if and only if L = L(G), where G is a regular grammar.

*Definition 12.* A scattered context grammar, G, is a quadruple (V, T, P, S), where V is a finite set of symbols, $T \subset V$, $S \in V \setminus T$, and P is a set of production rules of the form $(A_1, . . . , A_n) \rightarrow (w_1, . . . , w_n)$, $n \geq 1$, $\forall A_i : Ai \in V \setminus T$, $\forall w_i : w_i \in V^+$.

*Note:* A scattered context grammar combines context-free productions to create a context rewriting. Nevertheless, the context is not as tight as in the case of context-sensitive grammars.

Every category of the grammar defines a set of languages described by all grammars of a particular type.

## BASIC IDEAS

Firstly, we mention some basic ideas and expectations of our approach.

Let us consider some more complex process (regular car repair service performed periodically) consisting of a few sub-processes, see an example shown bellow.

### An example: Automobile Maintenance

A car passes through 4 basic departments (blocks of processes). At the beginning of each of them, there is an initial revision of the inspected car part.

1. Auto body
2. Car engine
3. Car electronics
4. Car chassis

Each of blocks can be divided into individual sub-processes/activities, e.g. car chassis service consists of: suspension service, wheel position service, wheel suspension service, etc.
A way of processes realization is changeable, e.g. which parties are including, what sequences etc. (arbitrary way, but there are certain limitations).

*Context limitations*

Interference with a certain car part can force another type of repair/revision, e.g. car chassis disassembly implies a necessity of wheel geometry check over to follow a correct order of operations. Moreover, geometry check must be done in such a case always. Performing this operation before car chassis disassembly is useless. There are a few of such limitations, thus, it is necessary to add them into our car repair model, too.

For example, we have a problem: workers repair a car and we need to have some feed-back (verification of the fact, that they have done all sub-processes and, moreover, they did it in a correct order). We gain a recipe of a way and order the operations were performed. Then, PC verifies if it is one of the correct possible ways (which workers chose).

We use scattered context grammars to model such a processes. The total number of operations in the car service is a finite number. Even number of mutual combination of these operations is a finite number. Thus, regular languages could be used to describe such a situation (Type-3 grammars of Chomsky hierarchy generating regular languages; every finite language is regular one).

On the other hand, in case of regular languages, it is necessary to make a listing of all possibilities, which can come on and this is very impractical (for the particular case listed below it is more than 6 millions of possibilities at all, context limitations decrease the size to more than half of million or more than 7 hundred thousands according to context setup). That is why we chose scattered context grammars even if we know that generated language is finite. An input of the program is not directly the grammar, but a simplified description of all process activities. There are the main independence groups–processes including a lot of sub-processes (sections). Such sub-processes can be realizes based on necessity of such a service at all or it is forced due to another service, which implies performance of another service.

The sequence of operations in the sections is again arbitrary (just the entrance operation is required to be performed before others, e.g. revision as the checkpoint), but activities within a certain operation must be completed in a correct order. Thus, we get several levels of independency:

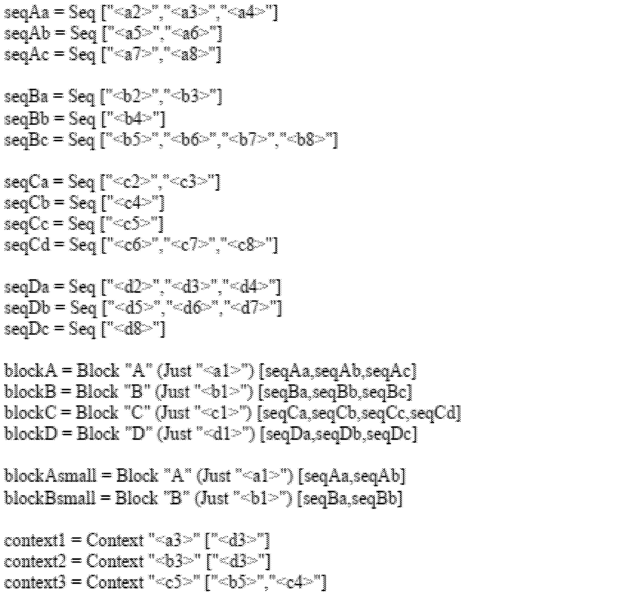
- blocks – e.g. body, engine, etc.
- operation sequences – e.g. wheel geometry setup
- activities within operations – elementary activities within operation, they must be performed in a particular order

The freedom of activities is bound by context rules that represent a requirement of one operation/activity if the other is performed (modelling of situations, when certain service requires another due to natural requirement given by car assembly).

### DEMONSTRATION

We have 4 basic processes (blocks of activities) and 4 workrooms in a car maintenance service. In each block, there is one fixed activity, e.g. inspection of the given car section. There are 3-4 sub-processes. Each of those sub-processes consists of activities set (1-4). We define 3 context bindings: there are 2 simple links (one activity implies another activity, thus, on sub-process implies another one) and 1 more complicated linking with more dependencies–in this case, we can take into account two possibilities, one activity implies another two and these two activities can be performed in an arbitrary order, or, the activities must be, in such a case, ordered two. In our example, one activity implies two others activities, e.g. proper sequence of activities (dismantling of front wheels together with repair → axle geometry/steering geometry etc.).

An example of the code in language Haskell, which describes processes definition and their dependencies is shown bellow:



```
seqAa = Seq ["<a2>","<a3>","<a4>"]
seqAb = Seq ["<a5>","<a6>"]
seqAc = Seq ["<a7>","<a8>"]

seqBa = Seq ["<b2>","<b3>"]
seqBb = Seq ["<b4>"]
seqBc = Seq ["<b5>","<b6>","<b7>","<b8>"]

seqCa = Seq ["<c2>","<c3>"]
seqCb = Seq ["<c4>"]
seqCc = Seq ["<c5>"]
seqCd = Seq ["<c6>","<c7>","<c8>"]

seqDa = Seq ["<d2>","<d3>","<d4>"]
seqDb = Seq ["<d5>","<d6>","<d7>"]
seqDc = Seq ["<d8>"]

blockA = Block "A" (Just "<a1>") [seqAa,seqAb,seqAc]
blockB = Block "B" (Just "<b1>") [seqBa,seqBb,seqBc]
blockC = Block "C" (Just "<c1>") [seqCa,seqCb,seqCc,seqCd]
blockD = Block "D" (Just "<d1>") [seqDa,seqDb,seqDc]

blockAsmall = Block "A" (Just "<a1>") [seqAa,seqAb]
blockBsmall = Block "B" (Just "<b1>") [seqBa,seqBb]

context1 = Context "<a3>" ["<d3>"]
context2 = Context "<b3>" ["<d3>"]
context3 = Context "<c5>" ["<b5>","<c4>"]

proc = PD "CarReview" [blockA,blockB,blockC,blockD] [context1,context2,context3]
```

*Context description*

The first activity forces another activity (in another place) or activities named in a list put behind the first, forcing, activity. For the present, there is a limitation that activities in context cannot be cyclic and/or chain. But, we can replace it by introduction of "pseudo-activity" application, which prevent from it. Moreover, cyclic dependence would mean impossibility of realization of the process in practice.

The whole description is very compact (code in Haskell) - generated scattered context grammar contains a bit more than one hundred of grammar rules. On the basis of it we verify individual activities, their order of appearance in the car reparation service, respectively.

*Future work*

Process is not only one, but there are a lot of similar processes (sub-processes). The main questions: How (optimal sequence of activities), When (duration and deadlines), Who (optimal resources scheduling). We consider longer time period, where a sub-process variability is not so expressive, but we need exact planning. We have obvious process, activities, duration, but many similar workrooms. We would like to achieve the optimal planning and scheduling, including an ability of total time estimate determination.

The main goal of this approach is the backward analysis, too, generating the grammar form, which satisfies conditions for LL-scattered grammars. There is suitable and effective analyzer available, see (Kolář 2008).

## CONCLUSION

In this paper, we have presented a new way of looking at analysis, planning and scheduling of complex systems – processes/sub-processes (e.g. auto repair). Planning and scheduling of a process should be done in a language, which is able to describe all important matter, relations and dependences among individual processes (it attains to get an optimized solution in respect to changeable conditions of the modelled process). We use Haskell as the main language, which enable us much more functionalities for our purpose as others languages (simple diction and programming of all ultimate facts) We demonstrate our approach on the Car repair example, where four basic processes with other sub-processes/activities are described and optimized in respect to contextual relations.

Additional research also needs to be conducted to investigate the scalability of this approach to larger systems (real processes with different characteristics, including many sub-processes etc.). We can study different situations and combinations of those sub-processes, too, that can appear during the main process development.

In the future, we suppose that the proposed model will be used for process planning support and will facilitate the whole process realization by controlling all important sub-processes and their parties.

## ACKNOWLEDGEMENTS

This research was supported by the Research Plan No. MSM 0021630528, Security-Oriented Research in Information Technology.

## REFERENCES

Anbari, F. 2005. *Q & As for the PMBOK Guide*. Project Management Institute. ISBN
1930699395

Billows, D. 2005. *Project Management Best Practices*

Kolář, D.: *Scattered Context Grammars Parsers*, In: Proceedings of the 14th International Congress of Cybernetics and Systems of WOCS, Wroclaw, PL, PWR WROC, 2008, pp. 491-500, ISBN 978-83-7493-400-8

Meduna, A.: *Automata and Languages: Theory and Applications*. Springer, London, 2000.

Fischer, L. (ed.): *BPM and Workflow Handbook*, Future Strategies Inc., 2007. ISBN 978-0-9777527-1-3

Meduna, A., Techet, J.: *Scattered Context Grammars and their Applications*. Prague, 2009.

Meduna, A., Švec, M.: *Grammars with Context Conditions and Their Applications*. 2005.

Greibach, S., Hopcroft, J.*: Scattered context grammars*. 1969.

Fernau, H.: *Scattered Context Grammars with Regulation*. Ann. Univ. Bucharest, Math.-Informatics Series, vol. 45, pp. 41—49. 1996.

Fischer, L.: *Excellence in Practice, Volume V: Innovation and Excellence in Workflow and Business Process Management*, ISBN 0-9703509-5-3

Harrison-Broninski, K.: *Human Interactions: The Heart and Soul of Business Process Management*. ISBN 0-929652-44-4

Češka, M., Rábová, Z.: *Gramatiky a jazyky*, Technical University of Brno, 1988.

Becker, J., Kugeler, M., Rosemann, M. (eds.). *Process Management*. Springer, 2003.

Smart, P.A, Maddern, H. & Maull, R. S.: Understanding Business Process Management: implications for theory and practice, British Journal of Management. 2008.

Trischler, W. E.: Understanding and Applying Value-Added Assessment . ASQ Quality Press, 1996.

Sparx Systems: UML Tutorials, 2004.

Haskell [online] www.thehaskellco.com/

## AUTHOR BIOGRAPHY

ŠÁRKA KVĚTOŇOVÁ was born in 1981, Brno, Czech Republic. She has studied Economics and Management at the Brno University of Technology, Faculty of Business and Management, Brno, Czech Republic. This year, she finished Ph.D. studies at Brno University of Technology, Faculty of Information Technology, Brno, Czech Republic. Her main research interests are software engineering, processes and project management with focus on Petri Net techniques and their application for software projects management.

DUŠAN KOLÁŘ was born in Hradec Králové, Czech Republic, and went to Brno University of Technology, Czech Republic, where he studied computer science and cybernetics and obtained his degrees in 1994 and 1998. Since then, he has been working at the university, presently at the Faculty of Information Technology. His main research interests are formal languages and automata and formal models with focus on Petri Nets and reactive/triggered models.

# System engineering approach for safety management of complex systems

R. Guillerm, H. Demmou and N. Sadou
CNRS ; LAAS, 7 avenue du colonel Roche, F-31077 Toulouse, France
University of Toulouse ; UPS, INSA, INP, ISAE ; LAAS, F-31077 Toulouse, France
rguiller@laas.fr
demmou@laas.fr
SUPELEC - IETR, Avenue de la boulais, F-35511 Cesson-Sevigne
nabil.sadou@supelec.fr

**KEYWORDS**
System Engineering, Complex System, EIA-632, Safety, Requirements.

## ABSTRACT

This paper presents a system approach for safety management of complex system. System engineering which is an interdisciplinary field of engineering that focuses on how complex engineering projects should be designed and managed is the framework of the approach. It allows taking into account the safety requirements in system engineering process to facilitates traceability of these requirements throughout the life cycle of the system. Processes of EIA-632 system standard are used to guide the proposed approach.

### Introduction

The system engineering process becomes more critical as our systems increase in size and complexity. Important system-level properties, such as safety and security (1), must be built into the design of these systems from the beginning; they cannot be added on or simply measured afterward.
Systems are changed. These changes are stretching the limits of current safety engineering approaches and techniques. These changes are challenging both safety processes, methods and tools. They concern:

- Fast pace of technological change
- Changing Nature of Accidents
- New types of hazards
- Increasing complexity and coupling
- Decreasing tolerance for single accidents
- More complex relationships between humans and automation
- Changing regulatory and public views of safety

Rasmussen has argued that major accidents are often caused not by a coincidence of independent failures but instead reflect a systematic migration of organizational behavior to the boundaries of safe behavior under pressure toward cost-effectiveness in an aggressive, competitive environment (2).
Weaknesses of the current safety processes (figure 1) can be resumed in the following points (non exhaustive list):

- Safety analysis involve some degree of intrinsic uncertainty. So, there is a degree of subjectivity in the identification of safety issues.
- Different groups need to work with different views of the system (e.g. systems engineers view, safety engineers view). This is generally a benefit but it can be a weakness if the views are not consistent.
- Definition of the safety requirements and their formalization.
- Traceability of safety requirements.
- Existing / traditional safety analysis techniques are difficult to use on modern, complex systems.
- Textual description of failure modes is often too ambiguous.
- System models are developed in electronic form, but no use is made of this for Safety/ Reliability analysis. Ideally there should be a common repository of all requirements, design and safety information.

Some points are due to the absence of a safety global approach. Indeed, safety must be addressed as global property and safety requirements (3) must be formulated not only in the small but in the large.
For example, the Ariane 5 and Mars Polar Lander losses are examples of system accidents. In both of these accidents, the components did not fail in terms of not satisfying their specified requirements. The individual components operated exactly the way the designers had planned the problems arose in the unplanned or misunderstood effects of these component behaviors on the

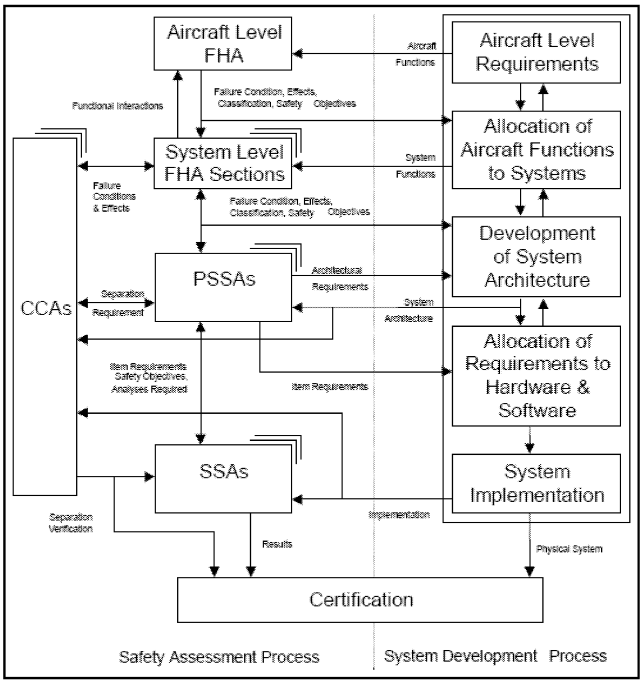

Aircraft Level FHA
Aircraft Functions
Aircraft Level Requirements
Functional Interactions
Failure Condition, Effects, Classification, Safety Objectives
Allocation of Aircraft Functions to Systems
System Level FHA Sections
System Functions
Failure Conditions & Effects
Failure Condition, Effects, Classification, Safety Objectives
Development of System Architecture
PSSAs
Architectural Requirements
CCAs
Separation Requirement
System Architecture
Allocation of Requirements to Hardware & Software
Item Requirements Safety Objectives, Analyses Required
Item Requirements
SSAs
System Implementation
Separation Verification
Implementation
Results
Physical System
Certification
Safety Assessment Process
System Development Process

Figure 1: Safety integration

system as a whole, that is, errors in the system design rather than the component design, including errors in allocating and tracing the system functions to the individual components. The solution, therefore, lies in systems engineering. A global approach is so necessary. Indeed, safety is clearly an emergent property of systems.

Some of these points are addressed by ESACS and ISAAC projects. ESACS project (4) developed a methodology and a platform that helps safety engineers automating certain phases of their work.

The ESACS platform can be used as a tool to assist the safety analysis process from the early phases of system design to the formal verification and safety assessment phases. It gives a partial response for the weakness cited above but it focused on the use of formal methods for safety assessment and does not propose a global approach do achieve it. For example, traceability of safety requirements and Human risk analysis are not considered.

ISSAC European project (5) which is the continuation of ESACS project proposes to take into account human errors analysis. It is achieved by injecting human errors in the formal model.

Nevertheless, these two projects are essentially concentred on formal method for safety assessment and not really in a global approach to achieve it.

ASSERT is another project, but, like above projects (ESACS and ISSAC), it focused on the method and tool. Moreover, only software failures are considered.

The norme IEC 1508 (6), (7) consider the overall lifecycle. It is considered as for the management of the safety throughout the entire life of the system, but it concerns only systems that require safety functions. It is guide for the implementation of the relevant safety functions. This work is part of a project in deploying System Engineering (SE)(11) (12) . We address the integration of safety management in system engineering process. The paper is structured into five remaining parts. The second part gives a brief introduction of the emerging discipline of system engineering in matter of key processes and the standard EIA-632 (13). The third part presents briefly the integration approach. In the forth part, an original approach for safety integration in system engineering process is proposed.

## The system engineering framework for complex system development

System Engineering is an interdisciplinary approach, which provides concepts that make it possible to build new applications. It is a collaborative and interdisciplinary process of problems resolution, supporting knowledge, methods and techniques resulting from the sciences and experiment. system engineering is a framework which helps to define the wanted system, which satisfies identified needs and is acceptable for the environment, while seeking to balance the overall economy of the solution on all the aspects of the problem in all the phases of the development and the life of the system. SE concepts are adequate specifically for complex problems; research issues undergone can bring a solution (11).

### System engineering concepts

System engineering is the application of scientific and engineering efforts in order to:

- Transform an operational need into a description of system performance parameters and a system configuration through an iterative process of definition, synthesis, analysis, design, test and evaluation.

- Integrate reliability, maintainability, availability, safety, survivability, human engineering and other factors into the overall engineering effort to meet cost, schedule, supportability and technical performance objectives.

System engineering is an interdisciplinary approach that:

1. Encompasses the scientific and engineering efforts related to development, manufacturing, verification, deployment, operations, support and disposal of systems products and processes.

2. Develops needed user trainings, equipments, procedures and data.

3. Establishes and maintains configuration management of the system.

4. Develops work breakdown structures and statements of work and provides information for management decision-making.

System engineering is a management methodology to assist designer through the formulation, analysis and interpretation of the impacts of proposed policies, controls or complete systems upon the need perspectives, institutional perspectives and value perspectives of stakeholders to issues under consideration.

System engineering is an appropriate combination of the methods and tools of a suitable methodological process and systems management procedures.

We distinguish three levels in System engineering as illustrated in figure 2. The first level, SE processes, focus on high-level issues, high-level requirements such as business needs and strategic needs and methods.

The second level, System engineering methodologies and methods, deals with all technical issues such as systems requirements design methodologies standards.

The third level, System engineering tools or technologies, covers the implementation issues concerning the tools to be used, the required technologies to respond to the various assets of requirements such as reliability, costs, maintainability and enabling technologies.

System engineering assists designer who desire to develop policies for management, direction, control and regulation activities relevant to forecasting, planning, development, production and operation of total systems Figure 2.

In System engineering best practice, we have the following chain:
Processes → Methods → Tools

These entities, such as processes, methods and tools, are the conceptual basis of our approach taken from System engineering best practice. In the first step, the processes can be identified with respect to the accumulated know-how, and can also be taken from a standard as the thirteen generic processes proposed in standard EIA-632. The second step concerns the methods to be used. The methods can be either developed or used by the existing one, which implement the process as we cannot choose a method for its flexibility or popularity but only if it reflects the semantics of the process. No taxonomy has yet been developed for corresponding processes and methods. The third step concerns the tools that do not correspond to the processes but the methods; hence in this approach we cannot use a tool to implement a process without first identifying the associated methods.

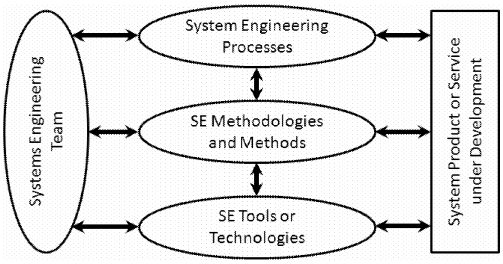

Systems Engineering Team
System Engineering Processes
SE Methodologies and Methods
SE Tools or Technologies
System Product or Service under Development

Figure 2: Three levels of system engineering.

### EIA-632 standard

One famous standard, currently used in the industrial and military fields, is the EIA-632. This standard covers the product life cycle from the needs capture to the transfer to the user. It gives a system engineering methodology trough 13 interacting processes grouped into 5 groups, covering the management issues, the supply/acquisition, design and requirement, realization and verification/validation processes. Figure 3 shows the interaction between all the 5 groups of processes, whose roles are (13):

1. Technical management processes (three processes): these processes monitor the whole process ranging from the initial idea of building a system until its delivery.

2. Acquisition and supply processes (two processes): these processes ensure the supply and acquisition (and are very close to logistics).

3. System design processes (two processes): these processes are on the elicitation and acquisition of requirements and their modelling, the definition of the solution and its design.

4. Product realization processes (two processes): these processes deal with the implementation issues of system design and its use.

5. Technical evaluation processes (four processes): these processes deal with verification, validation and testing issues.

Briefly, the operation of the proposed processes is:

- One acquisition request arrives and is treated by the supply process by establishing an agreement,

- The acquirer requirements are then transmitted to the System Design processes in charge of the elaboration of the logical solution, then the physical one, and also lots of sets of specified technical requirements, where each set is associated to a sub-system.

- The acquisition process is in charge to buy (if available in the market) or to make build the sub-systems responding to the different sets of specified requirements.
- Once the sub-systems received, the realization of the final product can begin, based on the design solution previously established and chosen.
- To finish, the final system will be transferred to the user, just after tests and final validation.

In parallel, all the previous processes are managed, rated and controlled by the technical management processes. And the technical evaluation processes allows to do system analysis (like risk analysis), requirement validations or system verification, during the development and when needed.
In fact, one or several sub-processes are defined for each 13 processes and the developer should decide which of the all 33 sub-processes apply. In this paper, only the system design processes and the technical evaluation processes are considered. These processes appear as the most important for safety management.

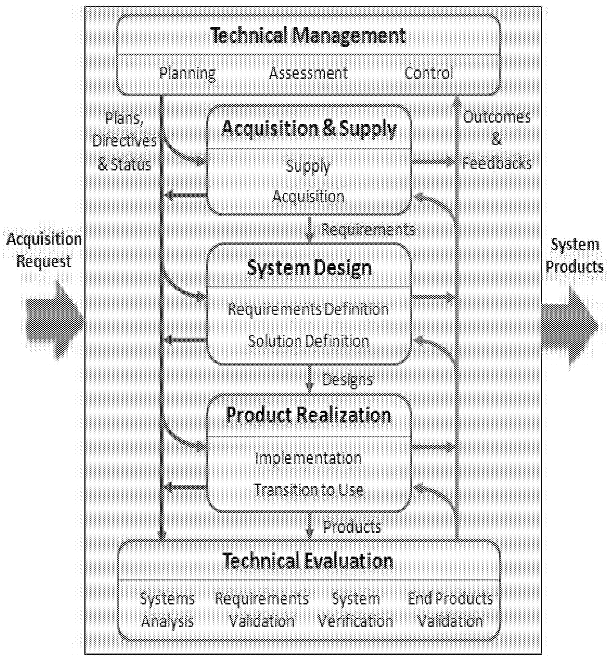


Figure 3: System engineering processes

## Safety in system engineering process

### Safety

Safety is an important system-level property, and must be built into the design of these systems from the beginning.

The safety assessment process can be decomposed into three main phases:

- preparation phase which initiates the assessment (Safety Target are defined)
- conduct phase in which the assessment is performed
- conclusion phase in which the assessment results are delivered.

System Engineering is the ideal framework for the design of complex system. In this work It is considered as a framework to manage safety.
A system engineering approach to safety starts with the basic assumption that the safety propriety, can only be treated adequately in their entirety, taking into account all variables and relating the social to the technical aspects (9). This basis for system engineering has been stated as the principle that a system is more than the sum of its parts.
The Safety management must follow all steps of SE From the requirements definition to the verification and the validation of the system. If we consider, for example, a reliability requirement defined for a global system, its formalization and analysis must allow ensuring that the technical solutions selected with design progression deals with this reliability requirement at sub-systems level and after their integration.
Note that this paper illustrates the proposed approach in term of process which must be defined independently to methods and/or tools (other projects which are focused on the methods and tools (4) and (5) for example). These different works will be exploited in the safety management from a global point of view.

### Integration approach

The integration of safety must concern all system engineering processes. This paper is focused only on:

- System Design processes,
- Technical Evaluation processes.

The safety requirements must be taken into account in requirements definition process. It allows the formulation, the definition, the formalization and the analysis of these requirements. Then a traceability (10) model must be build to ensure the taking into account of the requirements throughout the development cycle of the system.
These Safety requirements influence acquirer requirements, stakeholder requirements, system technical requirements, logical solution representations and physical solution representations.
Technical Evaluation processes define 12 types of sub-processes going from requirement statements validation to enabled product readiness. The sub-processes (The

task associated to each sub-process can be consulted in (13)) considered are:

- requirements statements validation,
- acquirer requirements validation,
- other stakeholder requirements validation,
- system technical requirements validation,
- logical solution representations validation,
- design solution verification.

The implementation of the approach consists in identifying and indicating in which way the safety must be considered for each sub-processes of EIA-632. In other words, the sub-processes of EIA-632 standard are translated or refined in terms of safety and included in system design process.

## EIA-632 sub-process to safety refinement

In this section we address EIA-632 processes with safety point of view.

### System design processes

The System Design Processes are used to convert agreed-upon requirements of the acquirer into a set of realizable products that satisfy acquirer and other stakeholder requirements.
Two processes are involved: the Requirements Definition Process and the Solution Definition Process. The relationship between these two processes is shown in Figure 4.

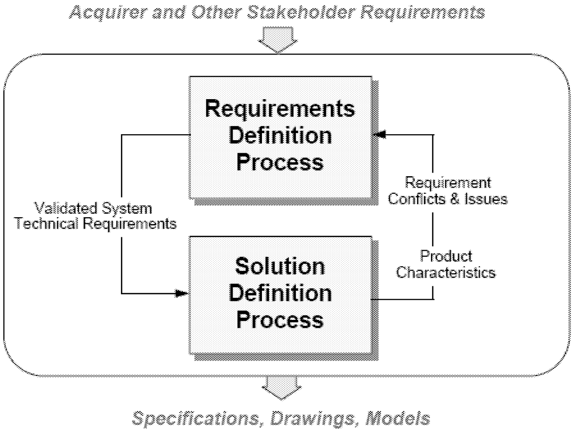

Acquirer and Other Stakeholder Requirements
Requirements Definition Process
Requirement Conflicts & Issues
Validated System Technical Requirements
Product Characteristics
Solution Definition Process
Specifications, Drawings, Models

Figure 4: System design processes

*Requirement definition process*
The goal of the requirements definition process is to transform the stakeholder (the acquirer and all other stakeholders who have an interest in the system) requirements into a set of technical requirements. For functional and no-functional requirements, if this distinction is not possible at the requirement elicitation process level, the analyzer may do it to categorize requirements. To perform this task, 3 sub-processes are associated with this process: the Acquirer Requirements, the Other Stakeholder Requirements and the System Technical Requirements sub-process. The Requirements Definition Process is re-accomplished, if necessary.

***a. Acquirer Requirements***
The developer shall define a validated set of acquirer requirements for the system, or portion thereof.
In the safety framework, acquirer requirements, generally, correspond to constraints in the system. It is necessary to identify and collect all constraints imposed by acquirer to obtain a dependable system. A hierarchical organization associates weight to safety requirements, following their criticality.
Some Standards are available to guide designer to define safety requirements. For example, for safety critical systems within the civil aerospace sector are developed subject to the recommendations outlined in ARP4754 (14) and ARP4761 (15). These standards give guidance on the 'determination' of requirements, including requirements capture, requirements types and derived requirements.
When the requirements are defined (16)some attributes can be used to facilitate their management.

***b. Other Stakeholder Requirements***
The developer shall define a validated set of other stakeholder requirements for the system, or portion thereof.
The same approach applied to acquirer requirements is applied to Other Stakeholder Requirements.

***c. System Technical Requirements***
The developer shall define a validated set of system technical requirements from the validated sets of acquirer requirements and other stakeholder requirements.
System technical requirements must be unambiguous, complete, consistent, achievable, verifiable, and necessary and sufficient for a system design.
For safety requirements, the system technical requirements traduce system performances. It consists on defining safety attributes ( Determine risk tolerability, MTBF, MTBR, failure rate for example).
Among all requirements formulated by the acquirer and other stakeholder, some of them are safety requirements. For example, critical events define reliability requirements in the sense that their taking into account must leads to a design of a system able to avoid these events.

*Solution Definition Process*

The Solution Definition Process is used to generate an acceptable design solution. This solution satisfies:

1. the system technical requirements resulting from the Requirements Definition Process,

2. the derived technical requirements from the Solution Definition Process.

Three sub-processes are associated with the Solution Definition Process.

***a. Logical Solution Representations***

The developer shall define one or more validated sets of logical solution representations that conform with the technical requirements of the system. In order to do this, he must in particular (1) do some tradeoff analysis, (2) identify and define interfaces, states and modes, timelines, and data and control flows, (3) analyze behaviors, and (4) analyze failure modes and define failure effects.

Formal models can be used for logical solution representations. The use of formal methods allows for automation of verification and analysis and for a tighter integration between system design and safety analysis. The model can automatically enriched with failures in order to perform safety analysis.

***b. Physical Solution Representations***

The developer shall define a preferred set of physical solution representations that agrees with the assigned logical solution representations, derived technical requirements, and system technical requirements.

The physical solution representations are derived from logical solution representation and must respects all requirements, particulary, safety requirements.

***c. Specified Requirements***

These requirements concern the design solution. The designer must ensure that the design solution is consistent with its source requirements. The safety analysis process allows the validation of these requirements.

**Technical Evaluation Processes**

The Technical Evaluation Processes are intended to be invoked by one of the other processes for engineering a system. Four processes are involved: Systems Analysis, Requirements Validation, System Verification and End Products Validation. The relationship between these processes is shown in Figure 5.

In this paper, we focus only on 3 processes of the technical evaluation:

1. Systems Analysis Process, which contains a Risk Analysis sub-process,

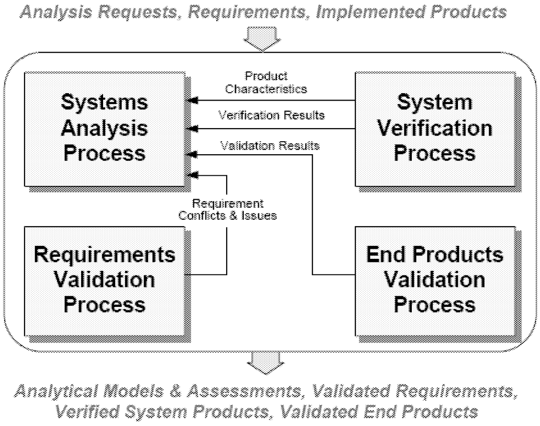

Analysis Requests, Requirements, Implemented Products
Systems Analysis Process
Product Characteristics
Verification Results
Validation Results
System Verification Process
Requirement Conflicts & Issues
Requirements Validation Process
End Products Validation Process
Analytical Models & Assessments, Validated Requirements, Verified System Products, Validated End Products

Figure 5: Technical evaluation processes.

2. Requirements Validation Process,

3. System Verification Process.

*Systems Analysis Process*

The Systems Analysis Process is used to:

1. Provide a rigorous basis for technical decision making, resolution of requirement conflicts, and assessment of alternative physical solutions;

2. Determine progress in satisfying system technical and derived technical requirements;

3. Support risk management;

4. Ensure that decisions are made only after evaluating cost, schedule, performance, and risk effects on the engineering or reengineering of the system.

***a. Risk Analysis sub-process***

The developer shall perform risk analysis to develop risk management strategies, support management of risks and support decision making.

Several techniques can be used to analyze risks, for example: fault tree, or Failure Mode, Effect, and Criticality Analysis. This step is very important, because it determines the risks of the system.

The step of risk analysis can generate safety requirements other than that defined by the acquirer and stakeholder. These new requirements must be taken into account.

*Requirements Validation Process*

Requirements Validation is critical to successful system product development and implementation. Requirements are validated when it is certain that they describe the input requirements and objectives such that the resulting system products can satisfy them. The Requirements Validation Process helps to ensure that the requirements are necessary and sufficient for creating design solutions appropriate to meet the exit criteria of the applicable engineering life cycle phase and of the enterprise-based life cycle phase in which the engineering or reengineering efforts occur. In this process, a great attention is done to traceability analysis, which allows verifying all the links among Acquirer and Other Stakeholder Requirements, Technical and Derived Technical Requirements, and Logical Solution Representations.

Like other requirements, safety requirements must be validated. The validation allows to design safe system. to facilitate this step, semi-formal solutions, like UML (17) or SysML (18) (which is an UML profile for systems engineering), can be used for good formulation of requirements. Indeed the diversity of people concerned by the system design project can have limited knowledge concerning the structure of a future system makes industry-scale requirement engineering projects so hard. So the UML or SysML with their different diagrams can be helpful.

*System Verification Process*

The System Verification Process is used to ascertain that:

1. The generated system design solution is consistent with its source requirements.
2. End products meet their specified requirements at each level of the system structure implementation (from the bottom up).
3. Enabling product development or procurement for each associated process is properly progressing.
4. Required enabling products will be ready and available when needed to perform.

Simulation is a good and current method used to achieve system verification. Other methods like virtual prototyping, model checking and other ones can be used.

## Conclusion

The approach presented in this paper concerns safety integration in system engineering process. It allows to give some guidelines to address efficiently safety of complex systems in all phase of system design. The approach is based on EIA-632 standard and can be resumed as follows:

- Elements of the approach: process → methods → tools
- System engineering processes handled using EIA-632 standard
- Integration of safety analysis in system engineering process

The paper addresses some processes of the standard EIA-632 independently to methods and/or tools. The processes addressed are system design process and technical evaluation process.

The work is in progress and other processes will be used and defined in the integration approach. The next step of the deploying system engineering project is to propose appropriate methods and tools for achieving each subprocesses of the EIA-632 standard.

## REFERENCES

[1] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing," IEEE Transactions on Dependable and Secure Computing, vol. 1, pp. 11-33, 2004.

[2] Jens Rasmussen. Risk Management in a Dynamic Society: A Modelling Problem. Safety Science, vol. 27, No. 2/3, Elsevier Science Ltd., 1997, pp. 183213.

[3] Finkelstein, A. (1993). Requirements Engineering: an overview. 2nd Asia-Pacific Software Engineering Conference (APSEC'93), Tokyo, Japan, 1993

[4] M. Bozzano, A. Cavallo, M. Cifaldi, L. Valacca, A. Villafiorita.Improving Safety Assessment of Complex Systems: An Industrial case study. FM 2003. Pisa, 8-14 September 2003

[5] O. Akerlund, P. Bieber, E. Boede, M. Bozzano, M. Bretschneider, C. Castel, A. Cavallo, M. Cifaldi, J. Gauthier, A. Griffault, O. Lisagor, A. Ldtke, S. Metge, C. Papadopoulos, T. Peikenkamp, L. Sagaspe, C. Seguin, H. Trivedi, L. Valacca. ISAAC, a framework for integrated safety analysis of functional, geometrical and human aspects. European Congress on Embedded Real-Time Software (ERTS 2006), Toulouse, 25, 26, 27/01/06

[6] Felix Redmill, Redmill Consultancy. An Introduction to the Safety Standard IEC 61508. Journal of the System Safety Society, Volume 35, No. 1, First Quarter 1999

[7] J. Brazendale. IEC 1508: Functional Safety: Safety-Related Systems. Software Engineering Standards Symposium, 1995.

[8] Richard C. Booten Jr. and Simon Ramo. The development of systems engineering. IEEE Transactions on Aerospace and Electronic Systems, AES-20(4):306309, July 1984.

[9] K. Kotovsky, J.R. Hayes, and H.A. Simon. Why are some problems hard? Evidence from Tower of Hanoi. Cognitive Psychology, vol. 17, 1985.

[10] Gotel, O. and Finkelstein, A. (1994). An Analysis of the Requirements Traceability Problem. 1st International Conference on Requirements Engineering (ICRE'94), Colorado Springs, April 1994, pp. 94-101.

[11] A.E.K Sahraoui, D. Buede, A. Sage, "issues in systems engineering research," *INCOSE congress*, Toulouse, 2004

[12] Systems Engineering Fundamentals. Defense Acquisition University Press, 2001

[13] EIA-632 : *processes for engineering systems.*

[14] Certification considerations for highly-integrated or complex aircraft systems. Society of Automotive Engineers, December 1994.

[15] Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment. Society of Automotive Engineers, August 1995.

[16] JL. Boulanger, Q-D. Van. A Requirement-based Methodology for Automotive Software Development , Int. Conf. on Modeling of Complex Systems And Environments, Ho Chi Minh City, Vietnam, July 07.

[17] G. Booch, J. Rumbaugh et I. Jacobson, *The Unified Modeling Language User Guide*, Addison- Wesley, 1998.

[18] SysML: Source Specification Project, http://www.sysml.org/

# ENGINEERING SIMULATION

# REDUCED SIMULATION´S MODEL OF A WHEEL LOADER BY USING THE BOND GRAPH TECHNIQUE TO USE IN TRAINING SIMULATORS.

G. Romero, J. Félez, J. Maroto, J. D. Sanz

E.T.S. Ingenieros Industriales, Universidad Politécnica de Madrid (UPM), C\ Jose Gutierrez Abascal 2, 28006, Madrid, Spain
E-mail addresses: {gregorio.romero, jesus.felez, joaquin.maroto, juandedios.sanz}@upm.es.

**KEYWORDS**
Bond Graph, earth moving machines, steering system, mechanism.

## ABSTRACT

This paper presents a model developed for simulating earth moving machines like wheel loaders. The developed model is used for real time simulation and is included in a full machinery simulator designated for the training..

The model includes a mechanical model of the chassis, axles, suspension systems, hydraulic actuators and mechanical models of the arms. All the models have been simulated using Bond Graph elements (Karnopp et al. 1990). The complete model has been developed as a modular system, using sub-models of each of the above-mentioned components. This approach helps to minimize both the number and complexity of the system equations obtained from the overall model.

Some simulation examples and results are also included.

## INTRODUCTION

Real time simulation is an indispensable requirement or models whose purpose is to test vehicle handling, since the driver expects to get an immediate response, as is the case in real life. Sometimes, when a vehicle's engineer decides to analyze the quality or safety, or to design controllers for ABS systems, normally rely on multibody dynamic models and each question requires a model of suitable complexity. The existing models span a wide range in the complexity spectrum. For instance, the design of a controller it might be enough to represent the vehicle as a point mass (Liang and Peng 1999); when we study suspensions we can use a quarter car model (Ando and Suzuki 1996); a half car model may be preferred when analyzing ABS performance (Alleyne 1997); finally, a full car and higher-order multibody models may be necessary (King and Ro 2002) for more advanced studies.

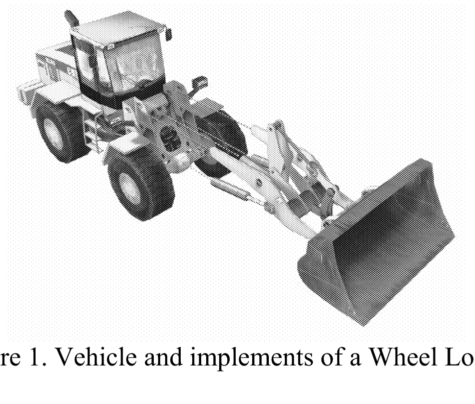


Figure 1. Vehicle and implements of a Wheel Loader

A framework for a modular approach to modeling 3D multibody systems is available in the literature (Pacejka 1985, Bos 1986, Ersal 2009) but there are not literatures about optimized earthmoving machines models in the items presented here.

When it comes to simulating machinery such as excavators or wheel loaders, the part corresponding to the vehicle's own dynamics is joined to the part related to the movement of implements, such as buckets, arms or actuators (fig. 1). Unlike traditional vehicles, these are lacking in suspension and need to incorporate an oscillating axle, located in the front or rear axle depending on the machine, so that the machine can be adapted to the unevenness of the terrain. In a traditional vehicle, it is the front wheels that turn the vehicle with no relative turn whatsoever being produced in the rear wheels with respect to the center of gravity. However, in the particular case of Wheel Loaders (fig. 2), all four wheels turn with respect to the center of gravity, which means that this phenomenon must be taken into account, as well as the specific way in which these machines produce the turn.

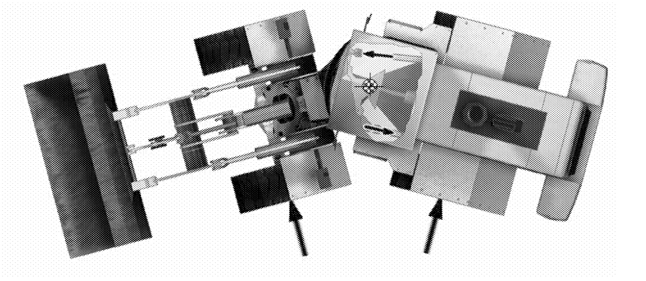


Figure 2. Turn in a Wheel Loader

As to the movements of the different implements, these are performed using the joint action of several hydraulic actuators and integrated mechanisms. These mechanisms are partially supported on the chassis of the machine. Since the main aim is to load, unload and move material using a scoop or a bucket, one of the objects of the simulation is to see the reaction occurring on the chassis of the machine when the different implements are being moved, as this involves a displacement of mass (Margolis and Shim 2002).

The aim of this paper is to demonstrate the validity of implementing kinematic or equivalent equations in cases where a dynamic simulation is not strictly necessary. The Bond Graph (Karnopp et al. 1990) technique enables systems belonging to the different areas of physics to be modelled in a way that is both intuitive and close to reality. It is a perfect technique for representing elements belonging to the area dealt with in this paper.

## STRUCTURE OF THE CHASSIS

In a Wheel Loader, the chassis is typically comprised of two front and rear parts joined in the middle by an axle which lets one part turn with respect to the other (fig. 2). The front part supports the front differential, arms and bucket, as well as the hydraulic actuators, while the rear part supports the cab, the rear differential, oscillating axle, engine, transmission and fluid tanks. The turning movement of the front part of the chassis with respect to the rear is performed by opening or closing the angle formed between both parts by activating two hydraulic cylinders, the turn being proportional to the angle turned by the steering wheel.

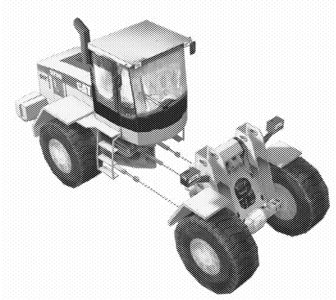


Figure 3. Wheel Loader chassis structure

As we have stated, unlike traditional vehicles, this type of machine incorporates an oscillating axle so that the machine can be adapted to the unevenness of the terrain (fig. 3).

On first inspection, the way to model the whole chassis would seem to be by introducing three rigid solids and then assembling them using the two axles. Since each of the rigid solids has its own reference system, a single global system must be worked with where all the velocities of the different points can be referenced, thereby making it possible to form the final set of equations (fig. 4).

Since the driver's cab is located over the rear chassis, it is the reference system of this part that must be used for calculating the velocities in the different points where the parts are joined.

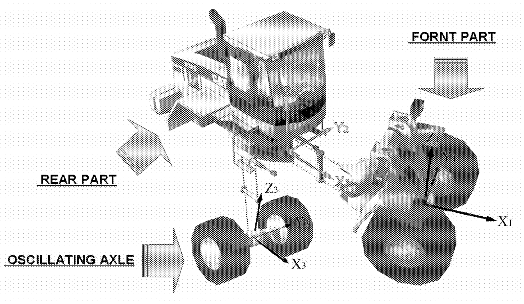


Figure 4. Chassis reference system

By observing the two joints where this change of reference needs to be made, it can be seen that in the first of these, a turn is made about the $Z_1$ axis parallel to Z, which means it would be sufficient to simply make a change on the $XY_1$ plane. Likewise, in the second joint, a turn is produced about the $X_3$ axis, which is parallel to X, it only being necessary to make the change of reference on the $YZ_3$ plane. Therefore, it is not necessary to work with three-dimensional coordinate transformations, but with planes, with all the simplification that this implies.

### Turning Movement

As is the case with a traditional vehicle with power steering, the power steering takes charge of changing the wheel directions with practically no effort by the driver, the pump has sufficient energy to change and maintain the direction of the wheels during the driving process. It is for this reason that it can be supposed that the angle formed between the front and rear part of the machine is proportional to the angle turned by the steering wheel.

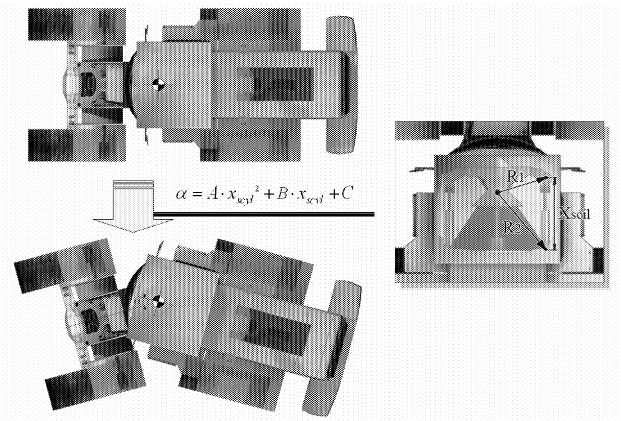


Figure 5. Re-positioning of the wheels in a Wheel Loader

If the positioning of each wheel is analyzed when a turn is being made, it can be seen how these relocate in accordance with the angle "α" (fig. 5) (max. ±40º) formed between the front and rear part of the machine; hence in accordance with the angle turned by the steering wheel. In each of the parts, the angle is distributed equally in both the front and the rear part.

Regarding the positioning of the wheels, it is possible to model a single rigid chassis instead of having to connect the two corresponding solids to the front and rear parts of the chassis and having to make a change of reference at its anchorage point. It will subsequently be necessary to change the position of the wheels on this chassis in respect of the centre of gravity, depending on the "α" angle which would be formed by the front and rear parts.

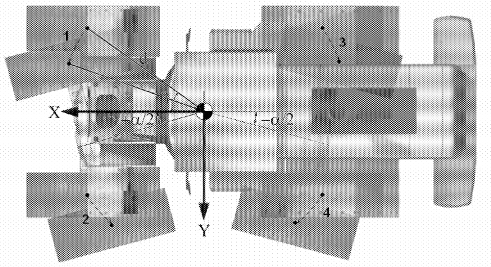


Figure 6. Re-positioning of wheel anchorage points

As a result of this change, a model with fewer equations is obtained, since the equations referring to one of the solids and to the change of reference have been eliminated. Thus, the points shown in the figure 6 corresponding to the wheels can be positioning depending on angle between front and rear parts "α" (table I):

| WHEEL | X | Y |
|---|---|---|
| - *(1) Right Front* | $+d\cdot\cos(\beta-\frac{\alpha}{2})$ | $-d\cdot\sin(\beta-\frac{\alpha}{2})$ |
| - *(2) Left Front* | $+d\cdot\cos(\beta+\frac{\alpha}{2})$ | $+d\cdot\sin(\beta+\frac{\alpha}{2})$ |
| - *(3) Right Rear* | $-d\cdot\cos(\beta-\frac{\alpha}{2})$ | $-d\cdot\sin(\beta-\frac{\alpha}{2})$ |
| - *(4) Left Rear* | $-d\cdot\cos(\beta+\frac{\alpha}{2})$ | $+d\cdot\sin(\beta+\frac{\alpha}{2})$ |

Table I. Calculation of wheel anchorage points

where "d" is the distance from the anchorage point to the center of gravity and "β" the angle initially formed respect of the horizontal. It must be said, that subsequently, in each of the wheels, the engine or braking torque needs to be separated into components taken about the global X and Y axes of the chassis, which is why it is essential to have information on the three linear and two angular velocity components at the wheel anchorage points.

Fig. 7 shows the Bond-Graph model corresponding to a three-dimensional solid (Karnopp and Margolis. 1993, Asgari and Hrovat 1991), expressed in local coordinates, with the information about $V_x$, $V_y$, $V_z$, $\omega_x$ and $\omega_y$ contained in one of the wheel anchorage points located at one of the coordinates referring to the $X_1$, $Y_1$ and $Z_1$ center of gravity. As can be seen in the fig. 7, the effect of the machine's weight has been included into the different local axes, instead of implementing equivalent changes of reference using TF type elements.

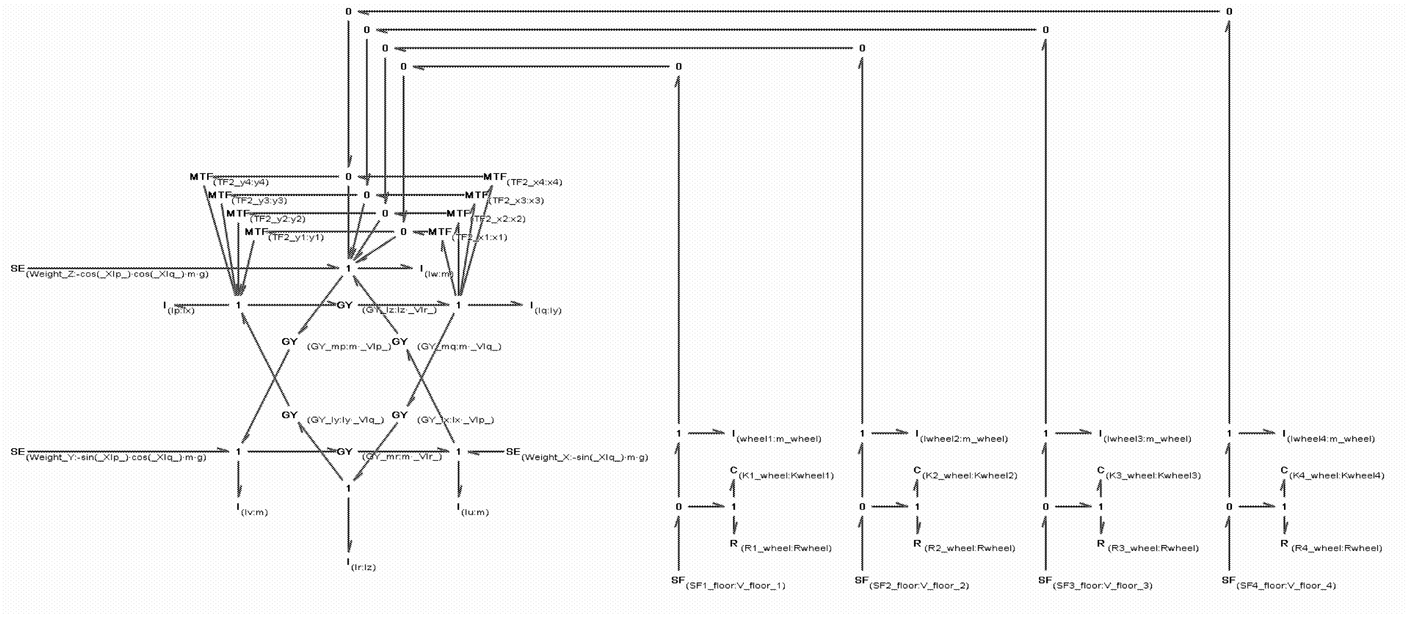


Figure 7. Chassis solid with anchorage points to the four wheels by means of a Bond-Graph.

## Movement Of The Oscillating Axis

As for the chassis, it is also essential to model the behavior of the oscillating axis with the object of adapting the machine to the unevenness of the terrain.

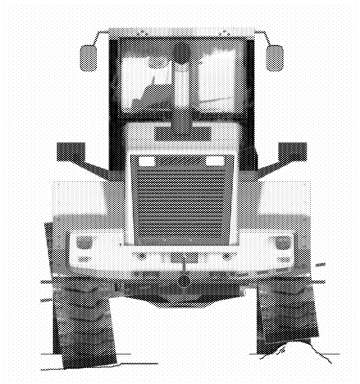


Figure 8. Movement of the oscillating axle

Firstly, it must be said that the maximum permitted movement is approximately since ±15°, the lateral movement of the wheels and all that entails, may be ignored, so that the vertical displacement of the wheels is all that is significant (fig.8).

Moreover, compared to the rest of the machine and due to the low maneuvering speed of these machines, the dynamic behavior of this axle may also be ignored.

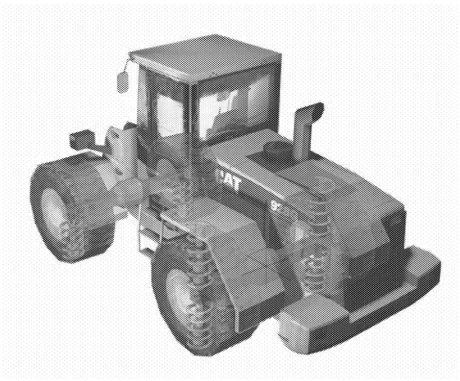


Figure 9. Dynamic model of the oscillating axle + suspension

However, it can be seen that due to the existence of the anchorage point, its movement is the average of that corresponding to the two wheels, and the movement of the suspension adjacent to one of the wheels is the opposite to that at the other (fig. 8 and fig. 9).

Fig. 7 showed the typical model for studying vehicle transversal dynamics where the suspension of each of the wheels can be seen as well as their inertias and rigidities. It can also be seen how the wheels has an excitation at its base, caused by a change in terrain (V_floor_1, 2, 3 or 4).

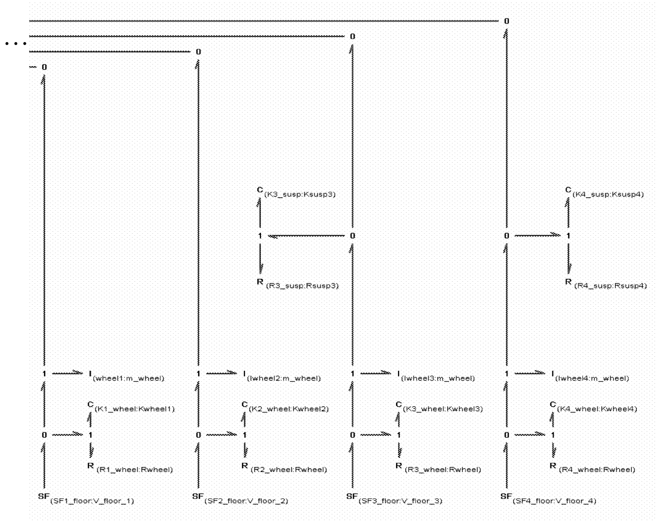


Figure 10. Suspension of a traditional vehicle (only rear part) by using the Bond Graph technique.

Thus, in the wheels corresponding to this axle, it will suffice to model a similar suspension to that of traditional vehicles (in the other wheels there is no suspension whatsoever), as is shown in figure 10, and then associate the movement at the point of anchorage as a combination of the movements of the wheels and the movement of the suspension with one another:

$$X_{left\_suspension} = -X_{right\_suspension} \qquad [1]$$

Fig. 11 shows the interrelation between the inertia velocities corresponding to each of the rear wheels of the machine ('Iwheel3' and 'Iwheel4', Right and Left wheels respectively) and the displacements existing in each of the suspension springs introduced (K3_susp and K4_susp) according to expressions [1].

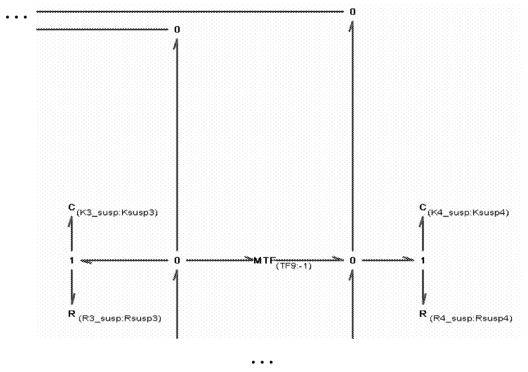


Figure 11. Modification of the rear suspension according with the oscillating axis expressions by Bond Graph.

If we apply a excitation at the base of one of the rear wheels caused by a change in terrain, such as can be seen in the graphs showing the results, the displacement of the suspension on one of the sides is the opposite to that of the other side (fig. 12).

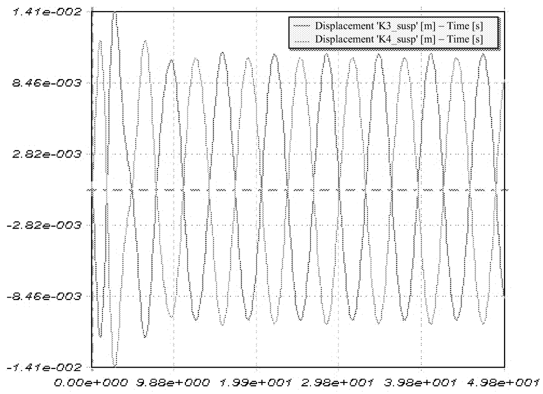


Figure 12. Displacements in the suspension

If we study the movement of the machine, the vertical velocity of the machine is the average of the movement of the wheels, such as we can see in next figure.

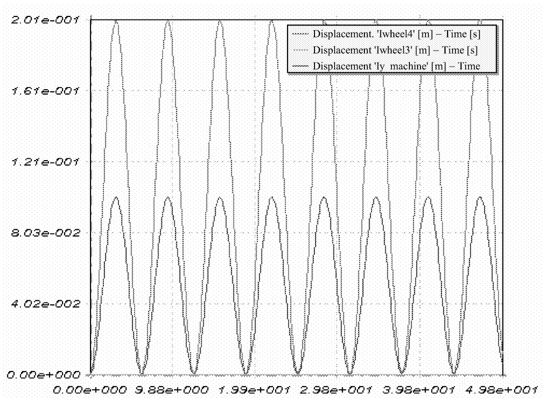


Figure 13. Vertical displacements of the machine's wheels

Fig. 14 shows that the difference between the two models is that the one with the oscillating axle has no displacement due to the vehicle's own weight, which is exactly what happens in reality.

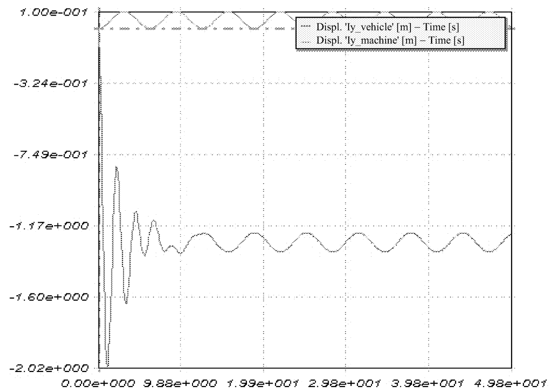


Figure 14. Displacements in the suspension

Under normal conditions, the values of the spring and damping of the suspension in the oscillating axle model will be small (just sufficient to quickly stabilize the machine), while they will be very great when the maximum turning angle of the oscillating axle is attained or superceded by ±15º.

## BUCKET HANDLING

The scoop or bucket is moved by using hydraulic actuators and one or several interlinked mechanisms (fig. 15). The functioning of these actuators is controlled by the machine operator handling the appropriate levers. This handling regulates the speed and direction in which the hydraulic actuators are required to work.

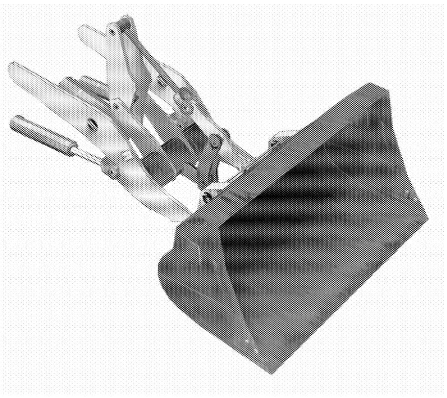


Figure 15. Implements of a Wheel Loader

Hydraulic systems modelling and simulation has been performed using a whole range of techniques, although currently a lot of territory remains to be researched in this area of Engineering. Traditionally, algebraic differential equations have been proposed based on the corresponding laws of physics (Cobo et al. 1998.). They have then been solved in various environments like MATLAB, Simulink, and MAPLE, to name but a few.

One of the drawbacks of this procedure lies in the fact that obtaining algebraic-differential equations is usually complex as well as the procedure for solving them, and on many occasions so much time is spent that real time simulation cannot be performed while costs also increase significantly.

Another possible option is to generate the model from zero using specific software, either by the finite elements method, using block diagrams, or using graphic techniques. These types of simulations are often oriented towards specific applications, frequently technological ones, and are therefore mainly focused on obtaining graphic or numerical results (Hydro+Pneu, OHC-Sim, HOPSAN, LVSIM) and move away from obtaining equations for the model.

Normally, in these machines, since the hydraulic circuit pump has enough power and there are intermediate hydropneumatic accumulators, the working velocity of the hydraulic actuators remains more or less constant for a constant lever position, and only varies as a function of the load in the bucket.

For this reason, we may assume that the velocity in the actuator pistons is proportional to the position of the control levers and the load in the bucket. To know the values of this velocity, it is necessary to simulate the hydraulic circuit and obtain the law of pressure and forces appearing in the piston, which considerably simplifies the complexity of this subsystem's model (Romero et al. 2008).

The mechanisms present in machines such as Backhoes and Wheel Loaders work on a single plane, which means a planar model can be studied instead of a three-dimensional one.

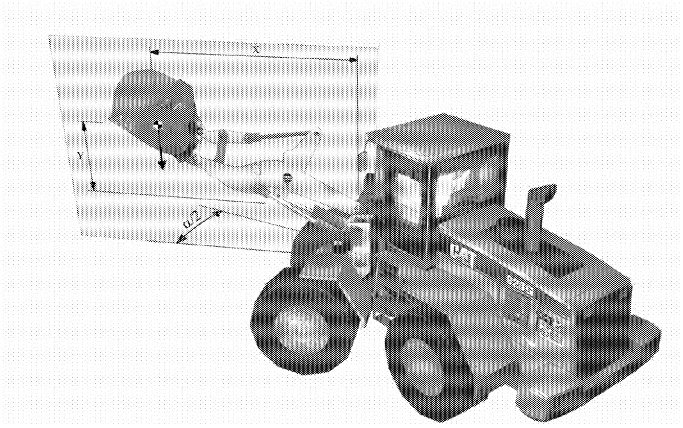


Figure 16. Reaction of the arm on the machine chassis

In order to be able to simulate the different implements, it is essential to develop a model of each hydraulic actuator, and assemble the different bars and actuators so as to obtain the equivalent mechanism (Romero et al. 2006; Romero et al. 2009) and incorporate it into the machine chassis so that the relevant actions and reactions will be produced on the chassis.

### Reactions On The Machine Chassis

As a general rule, it may be stated that the most important reaction that takes place in the machine chassis is that caused by the displacement of the load in the bucket, since it is the only mass that varies, either in volume or density, and to which the driver must be accustomed. Thus, it is essential to drive a Wheel Loader with the load as low as possible and avoid sudden braking, since this can cause the machine to overturn.

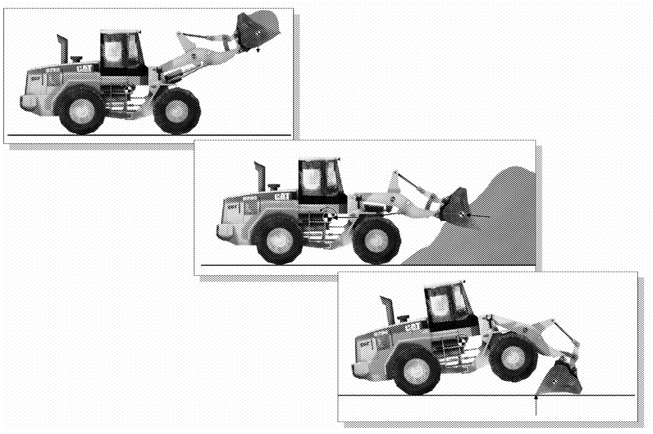


Figure 17. Reactions of the arm on the machine chassis

However, instead of anchoring the planar mechanism to some points located on the three-dimensional solid and thus, work with a single model where the reactions themselves would act directly on the chassis, we have preferred to isolate the mechanism.

So, when its kinematic simulation has been obtained, not only the angles in each of the arms can be obtained, but also the position of the point where the load acts.

When the coordinates of this point have been obtained, the resulting torque on the chassis can be quickly calculated, bearing in mind that the load at this point acts vertically and that the chassis may have a different orientation from the horizontal plane.

In this way, in order to obtain the behaviour caused by the reaction of the load on the chassis, it will only be necessary to introduce a point and a force into the three-dimensional solid.

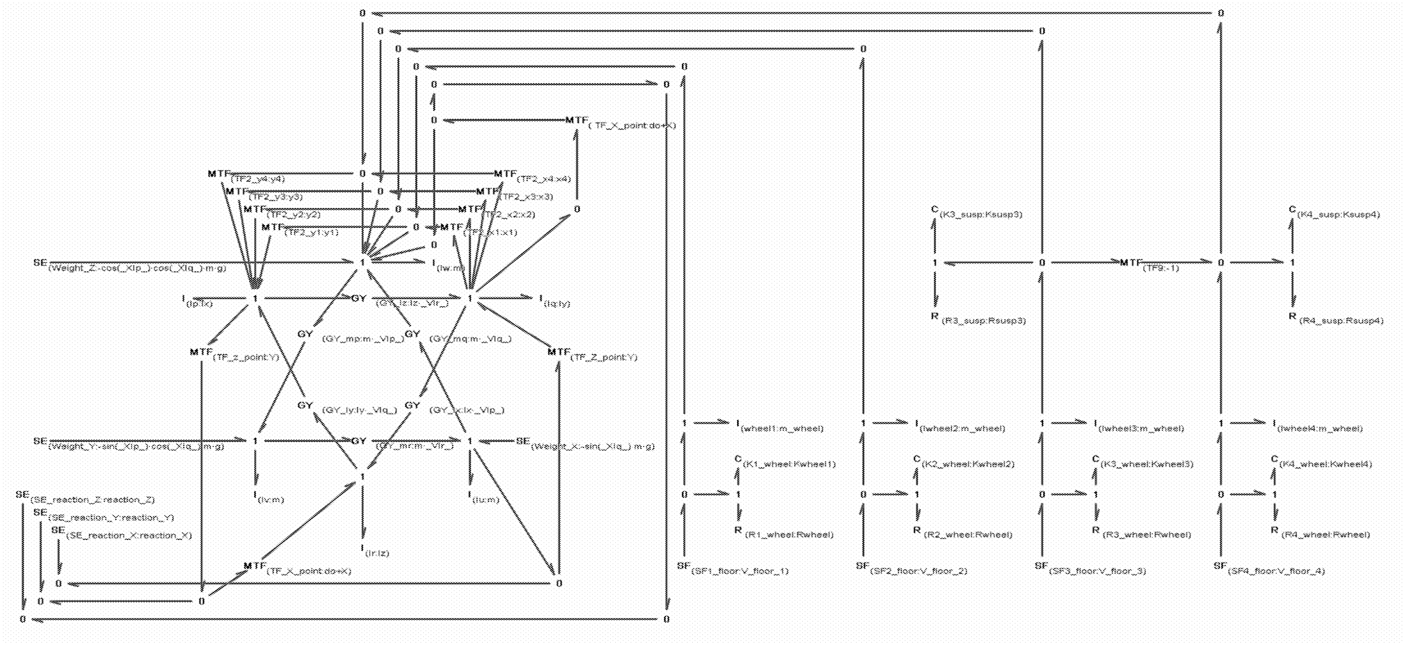


Figure 18. Chassis solid with anchorage points to the four wheels and one reaction forces by means of a Bond-Graph.

In fig. 18 it can be seen how the existence of a vertical load located at a point of local coordinates (do+X,Y,0) influences a three-dimensional solid. The coordinates of this point will constantly vary as the arm is extended or retracted and will be the result of the kinematic study of the arm.

The force will correspond to the weight of the load in the bucket, the reaction when the bucket load material or the reaction with the floor, once it has been resolved into the local coordinate axes of the chassis of the chassis.

Therefore, if we move the bucket away from the machine and unload the content of bucket to reduce the force due to it a few seconds after starting the simulation, the chassis of the machine undergoes a slight initial pitching until it stabilizes and then gradually acquires a different angle and the deformation of the wheel's, rear or front, are different (fig. 19).

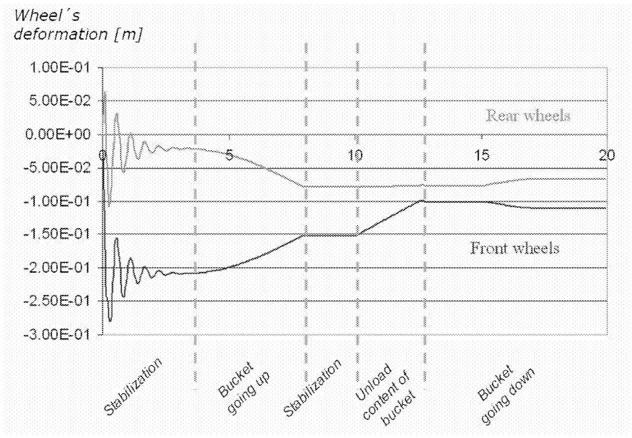


Figure 19. Wheel´s deformation

## CONCLUSIONS

As a final conclusion, it may be stated that in general it is possible to translate the results due to the action of external elements such as servopumps or hydraulic actuators to a model by using expressions according to the different parameters being handled (steering wheel, pedals, levers,...) or through the resolution of kinematic models (mechanisms) instead of trying to group everything together in a single model.

Also, the end model would only be left with those elements that it would be impossible to separate because they interact, such as the wheels, for instance, not dealt with here. On the other hand, we have seen the advantage of dealing with elements such as implement bars and actuators using a Bond-Graph approach.

It has thus been seen how a simulation model can be created starting out from simplifications and subdividing a large model into smaller ones.

It has also been seen how it is not necessary to simulate the dynamics associated with a mechanism at every instant, but only in cases where this information is really needed.

Finally, it should be pointed out that the equations so obtained are not only fewer in number but also in complexity, making a real time simulation possible by having reduced the time needed to perform it.

As an professional application of the developed model, it has been used to create a training simulator for a spanish simulation company.

## REFERENCES

Alleyne, A. 1997. "Improved Vehicle Performance Using Combined Suspension and Braking Forces". Vehicle System Dynamics, Vol. 27, Is. 4, pp. 235-265.

Ando, Y. and Suzuki, M. 1996. "Control of Active Suspension Systems Using the Singular Perturbation Method". Control Engineering Practice, Vol. 4, Is. 3, pp. 287-293.

Asgari, J. and Hrovat, D. 1991. "Bond graph models of vehicle 2D ride and handling dynamics" Proc. Of the 1991 ASME Winter Annual Meeting.

Bos, A. M. 1986. "Modelling Multibody Systems in Terms of Multibond Graphs with Application to a Motorcycle", Ph.D. Dissertation, University of Twente, Enschede, Netherlands.

Cobo, M., Ingram, R., Cetinkunt, S. 1998. "Modeling, Identification and real-time control of bucket hydraulic system for a Wheel type loader earth moving equipment". Journal Mechatronics, Vol. 8, pp. 863-885.

Ersal, T., Kittirungsi, B., Fathy, H.K., Stein, J.L. 2009. "Model reduction in vehicle dynamics using importance analysis". Vehicle System Dynamics, Vol. 47, Is. 7, pp. 851 – 865.

Karnopp, D.C., Margolis, D.L. and Rosemberg, R.C. 1990. "System Dynamics: A Unified Approach". John Wiley & Sons, Inc., Second edition.

Karnopp, D.C. and Margolis, D.L. 1993. "Analysis and simulation of planar mechanism systems using BG". J. of Mechanism Design. Vol. 101, Is.2, pp.187-191.

Kim, C. and Ro, P. I. 2002. "An Accurate Full Car Ride Model Using Model Reducing Techniques". Journal of Mechanical Design, Vol. 124, No. 4, pp. 697-705.

Liang, C.-Y. and Peng, H. 1999. "Optimal Adaptive Cruise Control with Guaranteed String Stability", Vehicle System Dynamics, Vol. 32, Is. 4, pp. 313-330.

Margolis, D. and Shim, T. 2002. "Instability Due to Interacting Hydraulic and Mechanical Dynamic in Backhoes", ASME International Symposium on "Advanced Vehicle Technologies".

Pacejka, H. B. 1985. "Modelling Complex Vehicle Systems Using Bond Graphs", Journal of the Franklin Institute, Vol. 319, Is. 1, pp. 67-81.

Romero, G., Félez, J., Martínez, M.L. and Maroto, J. 2006. "Kinematic analysis of mechanism by using Bond-graph language". Proc. of 2006 European Conference on Modeling and Simulation ECMS'06, pp. 193-202.

Romero, G., Félez, J., Martínez, M.L. and del Vas, J. J. 2008. "Simulation of the hydraulic circuit of a wheel loader by using the Bond Graph technique ".Proc. of 2008 European Conference on Modelling and Simulation ECMS'08, pp. 313 a 321.

Romero, G., Félez, J., Mera, J. M. and Maroto, J. 2009. "Efficient simulation of mechanism kinematics using bond graphs". Simulation Modelling Practice and Theory. Vol. 17, Is. 1, pp. 293-308.

## BIOGRAPHY

**GREGORIO ROMERO** received his Mechanical Engineering from the UNED (Spain) in 2000. He got his PhD Degree from the Technical University of Madrid in Spain in 2005 working on simulation and virtual reality, optimizing equations systems. He started as Assistant Professor at the Technical University of Madrid in Spain (UPM) in 2001 and became Associated Professor in 2008. He is developing his research in the field of simulation and virtual reality including simulation techniques based on bond graph methodology and virtual reality techniques to simulation in real time. He has published more than 35 technical papers and has been actively involved in over 20 research and development projects and different educational projects.

**JESÚS FÉLEZ** received his Mechanical Engineering and Doctoral degrees from the University of Zaragoza in 1985 and 1989. He started as Associate Professor at the Technical University of Madrid in Spain (UPM) in 1990 and became Full Professor in 1997. His main activities and research interests are mainly focused on the field of simulation, computer graphics and virtual reality. His research includes simulation techniques based on bond graph methodology and virtual reality techniques, mainly addressed towards the development of simulators. He has published over 50 technical papers and has been actively involved in over 25 research and development projects. He has served as thesis advisor for 30 master's theses and four doctoral dissertations.

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

**JUAN DE DIOS SANZ** received his Industrial Engineering (Electronic and Control) and Doctoral degrees from the Universidad Politecnica de Madrid, Spain, in 1992 and 2002. Since 1992 to 2000 worked at Detection and Protrection Train System Departmen inside the Transport Automation System division of ALCATEL in Spain, since 2000 he is the header of the Installation Area of the Research Centre on Railway Technologies (CITEF); being Assistant Professor at the Technical University of Madrid in Spain (UPM) since year 2006. His main activities and research interests are mainly focused on the field of railways, control train systems, passenger security and railway freight management. His main contribution is in the field of operational simulators and taking decission tools for planning, traffic scheduling and train control, as well as the relationship with normative as participant in national groups SCX9A and SCX9B of CENELEC or expert designated for IEC-62597 and new advances as technical advisory and member of the steering board of the Spanish Railway Technology Platform (PTFE). He has published over 20 technical papers and has been actively involved in over 40 research and development projects.

# ON LINE FAULT DIAGNOSIS OF A DIESEL ENGINE

Chady Nohra1
Hassan Noura2
Laboratoire des Sciences de l'Information et des Systèmes
Paul Cézanne University, Aix Marseille III
13397 Marseille Cedex 20, France
E-mail: 1cnohra@ndu.edu.lb
E-mail: 2hassan.noura@lsis.org

Rafic Younes
Mechanical Engineering Department
Lebanese University
Rafic Hariri Campus, Lebanon
E-mail: ryounes@ul.edu.lb

## INTRODUCTION

Long time considered pollutants, diesel engines are today as much clean as, if not cleaner than, the gasoline motors. In order to respect environmental standards, manufacturers implant systems of detection and fault-localization, called On-Board Diagnosis system (OBD), in order to enhance the engine effectiveness. Such systems became nowadays mandatory and have been embedded on the European diesel vehicles since 2003. Furthermore, OBD will soon be required for heavy trucks as well, both in the USA and the EU. This paper proposes a model-based diagnosis strategy of a diesel engine. Most previous researches in the field of diesel-engine diagnosis were conducted for a reduced number of parts of the diesel motor and did not consider a complete motor model. Furthermore, some of these methods have been based on statistical and experimental studies. Among these studies: the combustion diagnosis using neural networks (Yan and Ma. 2004), vibratory signals using Wavelet theory (Tafreshi et al. 2002), fuel injection faults using fuzzy logic (YongHe and LeiFeng 2004.)

Other studies involved model-based diagnosis for some specific parts in the Diesel engine such as faults in the cooling system of the diesel engine (Goh et al. 2002.), fault of the combustion process ( Lapuerta et al. 2005), and air-circuit faults (Nyberga and Stutteb. 2004) .

The strategy proposed in this paper aims at detecting, isolating, and estimating six faults in different parts of the diesel engine equipped with a variable-geometry turbocharger. Additionally, a complete model of the Diesel engine is adopted and an FDI system (Fault Detection & Isolation) based on the adaptive training theory of an on line non linear observer.

The paper is divided into 7 major parts. Part 2 shows the diesel engine model. Some comprehensive engine faults and their appearances in the system state-variables model are exploited in section 3. Section 4 discusses in detail the faulty diesel Model. Sections 5 and 6 study the fault diagnosis based on the theory of the adaptive training, discuss fault isolation and the architecture of an isolator. Section 7 exhibits the simulations that were carried out to validate our approach and which show effectiveness in the process of isolation of six faults on the diesel engine. Finally, we wrap up with a conclusion.

## DIESEL ENGINE MODEL

The Diesel Engine can be described by the following nonlinear system of equations and the block diagram of Fig. 1 below.

$$\begin{cases} \dfrac{d\vec{x}}{dt} = \xi(\vec{x}, \vec{u}, t) \\ y = \vec{x} \end{cases} \tag{1}$$

$\vec{x}$ : state vector, $\vec{u}$ : input vector, $t$ : time, $\xi$ is a non linear function, $y$ is the output vector which is, in this case, the state variables vector.

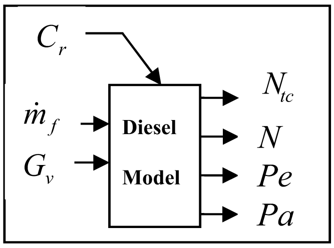


Figure 1. Diesel Engine Block Diagram

Vector $\vec{x}$ regroups motor rotational speed $w$ , turbocharger rotational speed $w_{tc}$ , admission and exhaust pressure $p_a$ and $p_e$ , respectively. Vector $\vec{u}$ contains fuel-oil mass flow rate $\dot{m}f$ ,, resistance load $C_r$ and variable geometry turbocharger $G_v$ .

The model used in this paper is the one developed by Xavier Doveefaz (X. Dovifaaz. 2001) . It can be expanded in the following way:

$$\begin{cases} I_{tc} w_{tc} \dfrac{dw_{tc}}{dt} = \left[ \eta_m \mathrm{P}_t \left( p_e, w_{tc}, GV \right) - \mathrm{P}_c \left( p_a, w_{tc} \right) \right] \\ Jw \dfrac{dw}{dt} = \left[ \eta_e \left( p_a, w, m_f \right) \dot{m}_f P_{ci} - pmf(w) \dfrac{Cy}{4\pi} w - C_r w \right] \\ V_e \dfrac{dp_e}{dt} = rT_e \left( p_a, w \right) \left[ m_a \left( p_e, w_{tc} \right) + m_f - \dot{m}_t \left( p_a, w_{tc} \right) \right] \\ V_a \dfrac{dp_a}{dt} = rT_a \left( p_a, w \right) \left[ \dot{m}_c \left( p_a, w_{tc} \right) - \dot{m}_a \left( p_a, w \right) \right] \end{cases} \tag{2}$$

$p_t$, $p_c$, $pmf$, $\eta_e$, $T_a$, $T_e$, $m_c$, $m_e$, $m_t$, *and* $m_a$ are the power of the turbine, power of the compressor, the average friction pressure in the motor, effective efficiency, the admission and exhaust temperatures, air flow rate in the compressor, exhaust, the turbine and the admission, respectively. These are semi-empiric nonlinear functions issued from experience (X. Dovifaaz. 2001).

The output is given by:

$$y = \left[w_{tc}, p_e, w, p_a\right] \tag{3}$$

Refer to Engine Specifications (R. Omran and R. Younès. 2007) for the Engine Characteristics and the notifications used for the foregoing parameters of the model.

## FAULTS AND MODELING

In order for the study to be more comprehensive and reflect as many frequent occurrences as possible, the following common faults will be considered in this paper:

### Fault n°1: Air Leakage in The Intake Chamber

Air leakage is modeled by the diameter of a hole in the intake chamber. As the leakage flow-rate varies according to the pressure, it can be modeled using the relation of Saint-Venant:

$$m_{leakage} = C_C \left(\pi \times \frac{d}{2}\right)^2 \frac{Pa}{\sqrt{r \times Ta}} \sqrt{\frac{2Cp}{r}} \sqrt{1 - \left(\frac{P_{atm}}{Pa}\right)^{(\gamma-1)/\gamma}} \tag{4}$$

Where d represents the diameter of the supposed hole, and $C_C$ represent the out-flow contraction factor.

### Fault n°2: Malfunctioning Of The Compressor

The compressor turns slowly because of a problem occurring at the group turbo-compressor level. The compressor flow rate will therefore be reduced; this fault can be modeled by multiplying the compressor flow rate by a constant lower than 1:

$$\dot{m}_C = (1 - K_C)\dot{m}_C \tag{5}$$

### Fault n°3: Fault in Opening The Intake Valves

It characterizes a bad opening of the admission valves; intake flow rate inhaled by the cylinders will be reduced:

$$\dot{m}_a = (1 - K_a)\dot{m}_a \tag{6}$$

### Fault n°4: Fault In The Intercooler

It represents a bad exchange of temperature; therefore the intercooler efficiency $\eta_{\text{intercooler}}$ will be reduced:

$$\eta_{\text{intercooler}} = (1 - K_{\text{intercooler}})\eta_{\text{intercooler}} \tag{7}$$

### Fault n°5: Fault In the Turbocharger Coupling

This fault type exhibits a coupling deterioration between the turbine and the compressor, therefore the coupling mechanical efficiency $\eta_m$ will be reduced and this is modeled the same way as for the previous fault:

$$\eta_{\text{m}} = (1 - K_{\text{t-c}})\eta_{\text{m}} \tag{8}$$

### Fault n°6: Fault in The Geometry of The Turbine

This is a fault in the turbocharger variable geometry; the geometry control coefficient $G_{\text{V}}$ will be reduced:

$$G_{\text{V}} = (1 - K_{G_{\text{V}}})G_{\text{V}} \tag{9}$$

## FAULTY DIESEL ENGINE MODEL

The diesel model equations (2) with faults integration can be put under the form:

$$\begin{aligned} \dot{x} &= \xi(x,u) + \varphi(x,u) + B(t-T)f(x,U,t) \\ y &= x \end{aligned} \tag{10}$$

Where $\boldsymbol{x} \in \Re^4$ is the state vector $x = [w_{tc}, pe, w, pa]$; $U \in \Re^9$ is the input vector that contains the principal inputs: u=[$m_f, Cr, GV$] and faults parameters input K = [$d, K_c, K_a, K_{GV}, K_{tc}, K_{ech}$]; therefore $U = [u, K]$; $\xi(x,u) \in \Re^4$ is the nominal model of the nonlinear system; This function regroups all differential equations of the diesel model (2). $\varphi(x,u) \in \Re^4$ represents the uncertainty or the model noise; $f(x,U,t) \in \Re^4$ is the fault function (Equations 4 to 9). The detection time of an event due to a fault is described by $B(t-T) = diag\{\beta_1(t-T_1),...,\beta_n(t-T_n)\}$, where $\beta_i(\tau)(i = 1,...,n)$ is a step function, i.e. the fault occurs at instant $T_i$ on the ith subsystem $(i = 1,...,n)$. $y$ is the output. Fault functions $f(x,U,t)$ are classified in two categories. The first contains: faults of air-leakage, admission, compression and of turbo compressor coupling. The task to explicit $f(x,U,t)$ in this case is simple. The second category contains the faults of intercooler and turbine variable geometry. It is impossible, for these faults, to respect the formalism 10 and it is then necessary to use neural networks to overcome this difficulty.

### Faults With Known Structure $f(x,U,t)$

- Hole in admission chamber: In this case, admission pressure is modified only in an explicit way by introduction of an air flow leakage (4).

Equation 2 becomes:

$$\frac{dp_a}{dt} = \frac{rT_a}{V_a}\left(\dot{m}_c - \dot{m}_a - \dot{m}_{fuite}\right)$$

$$f_{trou}(x,u) = \left[0;\ 0;\ 0;\ \frac{rT_a}{V_a}\dot{m}_{fuite}\right]^T \tag{11}$$

- Fault in admission valves: In this case the admission air mass flow is reduced $\dot{m}_a = \dot{m}_a(1 - K_a)$:

$$f_{Adm}(x,u) = [0;0;-\frac{rT_e}{V_e}\dot{m}_a K_a;\frac{rT_a}{V_a}\dot{m}_a K_a] \tag{12}$$

- Compression fault: For this fault $\dot{m}_c = (1 - K_c)\dot{m}_c$ is replaced in the admission and turbo-compressor equations:

$$f_{Comp}(x,u) = [\frac{1}{I_{tc}w_{tc}}\dot{m}_c.K_c.C_{pa}\cdot T_0\cdot\left(\pi_c^{\frac{\gamma-1}{\gamma}} - 1\right)\cdot\frac{1}{\eta_c};0;0;\frac{rT_a}{V_a}\dot{m}_c K_c]^T \tag{13}$$

- Fault in turbine-compressor coupling: the coupling efficiency $\eta_m = (1 - K_{t-c})\eta_m$ affects explicitly $w_{tc}$:

$$f_{tc}(x,u) = [-\frac{1}{I_{tc}w_{tc}}\left(\eta_m.K_{tc}.\dot{m}_e\cdot C_{pe}\cdot T_e\cdot\left(1 - \pi_t^{\frac{\gamma e-1}{\gamma e}}\right)\cdot\eta_t\right);0;0;0] \tag{14}$$

Thus, in case of known structure fault, figure 2 represents the fault formalism.

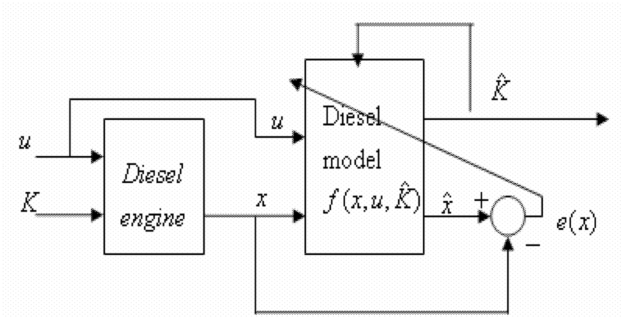


Figure 2: Parameters Estimation in the case of a Known Fault Function

### Faults With Unknown Structure $f(x,U,t)$

In this case, for example intercooler or variable geometry turbine, structure of the function $f(x,U,t)$ is unknown. It can be approached by an RBF neural network (Polycarpou and Helmicki 1995) which is a linear parametric approximation defined by ( Yan and Ma. 2004) :

$$y(X)=\sum_{i=1}^{7}\theta_i\Phi_i(X,c_i,\sigma_i) \tag{15}$$

Where $\boldsymbol{\theta_i}$ : weight vector associated with each of the basis functions $\Phi_i$ . X is the input vector; (in our case it contains, state variables $[w_{tc}pe,w,pa]$ and classic inputs $[mf,Cr,GV]$ ), $X=[x,u]=[w_{tc},pe,w,pa,mf,Cr,GV]^T$ .

$c_i=[c_{i1}\,c_{i2}\ldots c_{in}]^T$ are the center coordinates of the basis functions, and $\sigma_i$ denotes the widths of the basis functions. A very popular choice for the RBF is the Gaussian:

$$G_i(X,c_i,\sigma_i)=\exp\left(-\frac{1}{2}\left(\frac{(X_1-c_{i1})^2}{\sigma_{i1}^2}+\ldots+\frac{(X_n-c_{in})^2}{\sigma_{in}^2}\right)\right) \tag{16}$$

which can be used under its normalized form:

$$G_i(X,c_i,\sigma_i)=\frac{G_i}{\sum_{j=1}^{M}G_j}. \tag{17}$$

Due to this normalization, the neural network forms a partition of unity which improves the interpolation properties and makes the network less sensitive to the choice of the widths. Common algorithms for centers determination « $\boldsymbol{c_i}$ » are: "lattice method" and "clustering "(Werntges. 1993),or direct placement on the input data (Lippmann. 2002.), the last method was adopted while simulating diesel model for different values of fault parameter (for example, in the case of an intercooler fault: $K_{ech}=[0,0.15,0.3,0.45,0.6,0.75,0.9]$ for different values of the main inputs $u=[\dot{m}_f,Cr,GV]$ with the vectors of corresponding state $x=[w_{tc},pe,w,p_a]$ to be able to identify the corresponding centers $c_i=[w_{tc},pe,w,p_a,\dot{m}_f,Cr,GV]$ ). Standard deviations can then be chosen by the nearest neighbor method, i.e, proportional to the distance to the nearest neighbor center ( Werntges. 1993).

For on-line learning a training algorithm which can be applied during regular operation has been developed. For local tuning of the neuron weights, a simple computationally efficient learning algorithm, the normalized least-mean-squares (NLMS) rule as given by (Moody and Darken 1989).

$$\boldsymbol{\theta}_i^{new}=\boldsymbol{\theta}_i^{old}+\boldsymbol{\mu}.\boldsymbol{e}(\boldsymbol{x}).\frac{\boldsymbol{\Phi}_i(\boldsymbol{x})}{\sum_{j=1}^{M}\boldsymbol{\Phi}_j^2(\boldsymbol{x})} \tag{18}$$

$e(x)$ denote the states estimation error. $\mu$ represents the learning rate which must be within the range $0<\mu<2$ . for each fault different parameters values (reduction of the intercooler efficiency, Geometrie control coefficient) for different values of x and u were simulated and the correspondent vector $\hat{\theta}$ was calculated by the adaptive law (18).

## FAULT DETECTION AND ISOLATION

The Fault Detection and Isolation (FDI) problem has received considerable attention in the control systems design and there is a rich body of literature that deals with such applications (Isermann 1984; Gertler 1998). Model-based FDI techniques utilize output observers and are based on residual generation and evaluation which should reflect any change in the dynamics of the system. Fault detection is the first step for fault diagnosis followed by the critical task of fault isolation. A robust strategy to diagnose the fault of non-linear systems has been proposed by Demetriou and Polycarpou in 1998 (Demetriou and Polycarpou 1998). The basic idea was to use adaptive learning of an on line observer which estimated the amplitude of faults. Robust thresholds were given theoretically to guarantee no false alarm and the stability was proved by the Lyapunov method ( Li and Zhou 2004) . Once the fault is detected, a bank of non-linear adaptive observer is activated for the goal of isolation.

### Sliding Mode Adaptive Observer

Assumption 1: All the states of (10) are measurable.
Assumption 2: The model uncertainty is bounded, i.e. $|\varphi_i(x,u)|\le\overline{\varphi_i}$ Where $\varphi_i(x,u)$ is the ith element of $\varphi(x,u)$ $(i=1,\ldots,n)$ and $\overline{\varphi_i}$ is a given positive constant.

Under these assumptions, a sliding mode adaptive observer is constructed as follows,

$$\dot{\hat{x}}=-\Lambda\hat{x}+\Lambda x+\xi(x,u)+M(\tilde{x})+\hat{f}(x,u,\hat{\theta}) \tag{19}$$

Where $\Lambda=diag(\lambda_1,\ldots,\lambda_n),\lambda_i>0,(i=1,\ldots,n)$ . is to be designed with −Λ as the eigenvalue matrix of the observer;

$\hat{f}(x,u,\hat{\theta})=[\hat{f}_1(x,u,\hat{\theta}_1),\ldots,\hat{f}_n(x,u,\hat{\theta}_n)]^T$ is the online approximator, which is used to estimate the fault function with $\hat{\theta}_i\in\Re^{pi}$ as the parameters $(i=1,\ldots,n)$ $\tilde{x}:=x-\hat{x}=[\tilde{x}_1,\ldots,\tilde{x}_n]^T$ is the states estimation error; and $M(\tilde{x})=[M_1(\tilde{x}_1),\ldots,M_n(\tilde{x}_n)]^T$ is the sliding mode term with boundary layer control, which can avoid the “chattering” phenomenon effectively (Walcott and Zak 1988) .

The sliding mode term is constructed as follows,

$$M_i(\tilde{x}_i) = \begin{cases} \overline{\varphi}_i sign(\tilde{x}_i) & si \quad |\tilde{x}_i| > \eta_i \\ \overline{\varphi}_i \tilde{x}_i / \eta_i & si \quad |\tilde{x}_i| \leq \eta_i \end{cases} \qquad i = (1,\ldots,n) \tag{20}$$

Where $\eta_i > 0 \ (i = 1,\ldots,n)$ is the threshold of boundary layer, which is selected as a small positive constant and satisfies $\eta_i << \overline{\varphi}_i$.

*Theorem 1.* ( Li and Zhou 2004) .Assume $\tilde{x}_i(t_0) = 0$ (according to Assumption 1), then the estimation error of the ith state satisfies $|\tilde{x}_i| \leq \varepsilon_i \quad (i = 1,\ldots,n)$ before the fault occurs, where $\varepsilon_i = \min[(1/2)\sqrt{\eta_i \varphi_i / \lambda_i}, \eta_i]$.

*Remark:* Since it is usually satisfied that $\eta_i << \overline{\varphi}_i$ then we have $\varepsilon_i = \eta_i$ According to Theorem 1, the robust fault detection strategy is obtained intuitively:

when $|\tilde{x}_i| \leq \varepsilon_i$ for all $i = 1,\ldots,n$ there is no fault; at the first time $|\tilde{x}_i| > \varepsilon_i$, we claim that the ith subsystem is faulty and simultaneously we start the adaptive law, i.e. $\dot{\hat{\boldsymbol{\theta}}}_i \neq 0$.

The parameters of the fault functions can be approximated by using the adaptive law (23) written under the following form ( Li and Zhou 2004):

$$\dot{\hat{\theta}}_i = \begin{cases} \Gamma_i(-k_i\hat{\theta}_i + \Omega_i\tilde{x}_i) & \text{if} \quad |\tilde{x}_i| > \varepsilon_\mathrm{i} \\ 0 & if \quad |\tilde{x}_i| \leq \varepsilon_\mathrm{i} \end{cases} \tag{21}$$

$$\hat{\theta}_i(t_0) = 0 \quad (i = 1,\ldots,n)$$

Where $\boldsymbol{\Gamma}_i \in \Re^{pi \times pi} > 0$ is the learning speed matrix and $\boldsymbol{k}_i > 0$ is the feedback coefficient which is always selected as a small positive number.

*Theorem 2* (Stability) ( Li and Zhou 2004) : Let $\tilde{\theta}_i = \theta_i^* - \hat{\theta}_i$ be the estimation errors of parameters. Under the adaptive law (21), the estimation errors of states and parameters coming from (19) and (21) are all uniformly bounded.

## APPLICATION TO THE DIESEL ENGINE

Applying previous algorithms, we can detect and isolate majority of the abovementioned fault. Let's recall that the system of equation of the faulty diesel model has been put under the following form:

$$\dot{x} = \xi(x,u) + \varphi(x,u) + B(t-T)f(x,u,t)$$

$$x = [w_{tc}, p_e, w, p_a]\,; u = [\dot{m}_f, Cr, GV, d, K_f, K_c, K_a, K_{GV}, K_{ci}, K_{tc}, K_{ech}];$$

The function $\xi(x,u)$ groups all equations of the diesel model (2).

The function $f(x,u,t)$ groups parametric fault equations (11-15).

The function $\varphi(x,u)$ has been modeled as a Gaussian noise proportional to the admission pressure.

## SIMULATION

The engine is equipped with:

1- Sensors that measure the system states:

- of an admission pressure sensor ( $p_a$ )
- of a engine rotational speed sensor ( $w$ )
- of a turbocharger rotational speed sensor ( $w_{tc}$)
- of an exhaust pressure sensor ( $p_e$ )

2 – a fuel-oil air flow rate sensor ( $\dot{m}_f$ ) and the input to variable geometry control ( $GV$ )

- Resistant torque $C_r$ cannot be measured directly. It can be calculated according to equation (2) while using a new sensor of the admission air flow rate ( $\dot{m}_a$ ) as well as the available previous measures of the rotational speed (w) and the fuel-oil flow rate ( $\dot{m}_f$ ).

## FDI Structure

For every fault a block is constructed that estimates fault parameter by supposing that the perturbation signals of the state variables derive from the fault corresponding to the block (the function $f(x,u,t)$ will be equal to the expression of the corresponding fault). A bank of *N* observers is used in the proposed fault diagnosis scheme, where *N* is the number of parametric faults. Once a fault is detected, then all the *N* isolation observers estimate the corresponding fault parameter. A logic unit eliminates unacceptable parameter faults (for instance, a reduction factor greater than one or negative reduction factor, negative diameter). The structure of the FDI is given in the Fig. 3 below. Each estimator foresees the value of the corresponding fault parameter. If the fault doesn't correspond to the estimator fault, the parameter value must be zero or doesn't belong to parameter acceptable value. In this case fault is considered isolable. If two or several estimators give acceptable values for a given fault, then this fault cannot be isolated.

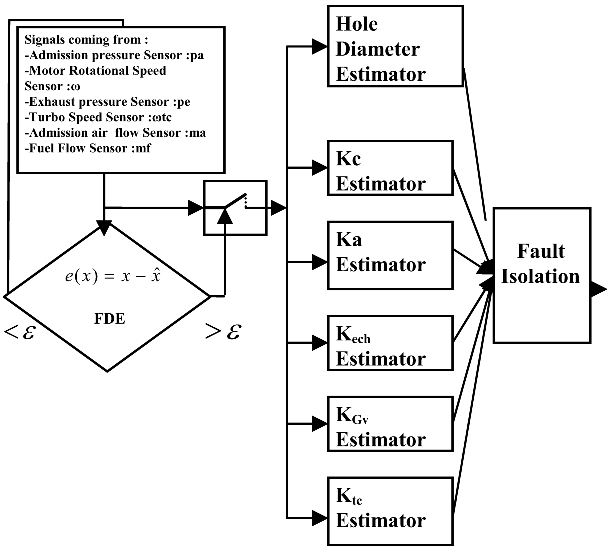


Figure 3: The Overall FDI Structure

Figure 4 shows the output of six estimators for the fault of a hole in the admission collector of diameter 10mm at t=1sc.

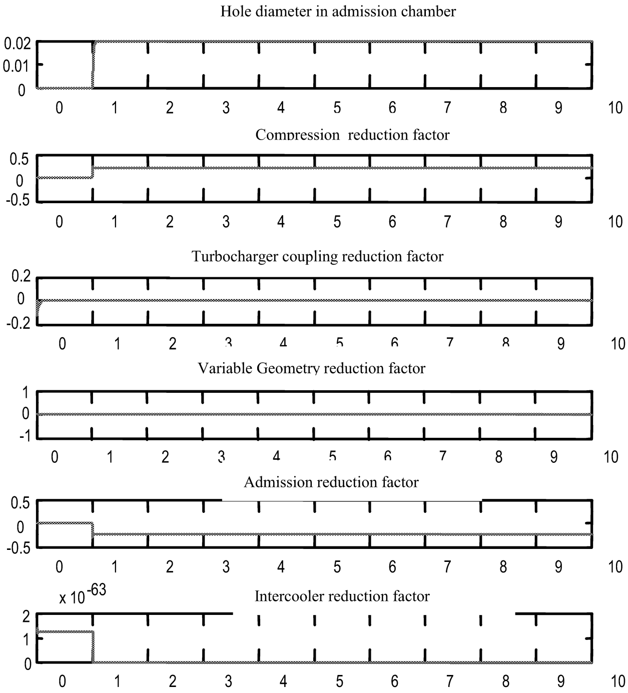
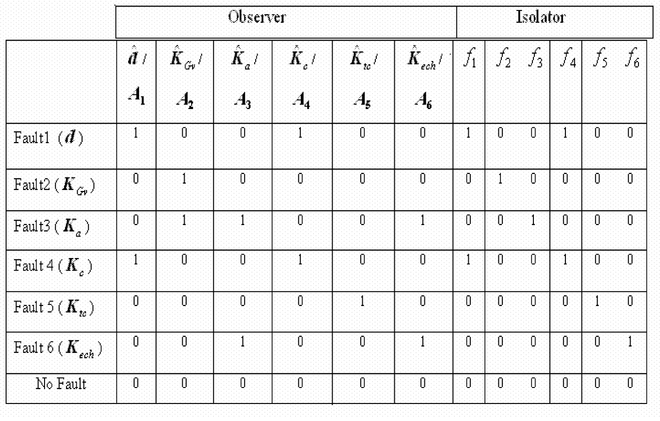


Figure 4: Output of the six Estimators at the time of an Admission Air-Leakage Fault of diameter 10mm

This figure shows that the parameters values of all faults are zeros or unacceptable (negative or bigger than 1) except the hole diameter in the admission chamber and the compressive reduction factor. This fault is thus isolable of all others faults except the compressor one. In order to improve the isolation, a set of logic variables Ai ( i=1…6 ) is associated to each estimator with Ai = 1 if the corresponding isolator estimates an acceptable value of the fault parameter (a value between 0 and 1 for a reduction factor , and a positive value for the diameter of the hole in the admission chamber) and zero otherwise.Table 1 summarizes the responses of all estimators to different faults and the values of the corresponding logic parametersAi.



| | Observer | | | | | | Isolator | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $\hat{d}$ / $A_1$ | $\hat{K}_{Gv}$ / $A_2$ | $\hat{K}_a$ / $A_3$ | $\hat{K}_c$ / $A_4$ | $\hat{K}_{tc}$ / $A_5$ | $\hat{K}_{ech}$ / $A_6$ | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ |
| Fault1 ($d$) | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Fault2 ($K_{Gv}$) | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| Fault3 ($K_a$) | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| Fault 4 ($K_c$) | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| Fault 5 ($K_{tc}$) | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| Fault 6 ($K_{ech}$) | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 |
| No Fault | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 1: Symptoms Of Different Faults On Different Estimators

A set of logic functions fi are defined to isolate the faults: $f_1 = A_1; f_2 = A_2.\overline{A}_3; f_3 = A_2.A_3; f_4 = A_4; f_5 = A_5; f_6 = A_6\overline{A}_2$ A s shown in table.1 all faults can be isolated except for fault1 (Air-leakage in admission chamber) and fault 4(compressive reduction fault).

## CONCLUSION

This work allowed us to enhance the importance of a complete model for the Diesel engine in the domain of fault diagnosis and isolation. The studied faults are representative of some frequent breakdowns on this type of engine as the deterioration of the transmission between the various mechanical components. The use of an on line observer with sliding mode allows to assure the functions of fault detection and isolation in a much reduced time. The indispensable measures for this method are, excluding the one of turbo-compressor, already available on the modern vehicle.

## REFERENCES

Chen, S., C.F.N. Cowan, and P.M. Grant. 1991. "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks," IEEE Transactions on Neural Networks, Vol. 2, No. 2, March 1991, pp. 302-309.

Demetriou, M. A., & Polycarpou, M. M. 1998. "Incipient fault diagnosis of dynamical systems using online approximators". IEEE Transactions on Automatic Control, 43, 1612–1617.

Gen-Ting Yan, Guang-Fuma .2004 .“Fault Diagnosis of diesel engine combustion system based on neural networks”

H. Werntges. 1993. “Partitions of unity improve neural function approximators,” Proc. IEEE Int. Conf. Neural Networks (ICNN’93), San Francisco, CA, vol. 2, pp. 914–918.

J. J. Gertler. 1998. "Fault Detection and Diagnosis in Engineering Systems", Marcel Dekker, New York, 1998.

Keng Boon Goh, Sarah K. Spurgeon, N. Barrie Jones.2002. “Fault diagnostics using sliding mode techniques” Control Engineering Practice 10 . 207–217

Linglai Li, Donghua Zhou. 2004. “Fast and robust fault diagnosis for a class of nonlinear systems: detectability analysis”. Computers and Chemical Engineering, pp 2635–2646.

Mattias Nyberga, Thomas Stutteb. 2004. “Model based diagnosis of the air path of an automotive diesel engine” Control Engineering Practice, Volume 12, Issue 5, 1 May 2004, Pages513-525

M. Lapuerta, O. Armas, J.J. Hernandez. 2005. “Diagnosis of DI Diesel combustion from in-cylinder pressure signal by estimation of mean thermodynamic properties of the gas”. Control Engineering Practice 13 , 189–203

Polycarpou, M. M., & Helmicki, A. J. 1995 .Automated fault detection and accommodation: a learning systems approach. IEEE Transactions on Systems, Man, and Cybernetics, 25, 1447–1458.

R. Isermann.1984. “Process fault detection based on modeling and estimation methods: a survey”, Automatica, vol. 20, pp. 387-404.

R. Lippmann. 2002. “An introduction to computing with neural nets,” IEEE Acoust., Speech, Signal Processing Mag., vol. 4, pp. 4–22, Apr.

R. Omran, R. Younès. 2007. “Genetic Algorithm for Dynamic Calibration of Engine's Actuators” SAE paper 2007-01-1079.

R.Tafreshi, H.Ahmadi, F.Sassani, G.Dumont. 2002. “Informative Wavelet Algorithm in Diesel Engine Diagnosis” IEEE International Symposium on intelligent Control Vancouver, Canada October 27-30.

Walcott, B. L., & Zak, S. H. 1987. "State observation of nonlinear uncertain dynamical systems". IEEE Transactions on Automatic Control, 32, 166–169.

X. Dovifaaz. 2001. “Modélisation et commande de moteur Diesel en vue de la réduction de ses émissions”. PHD thesis, UPJV, Amiens, France.

YongHe and LeiFeng. 2004. “Diesel Fuel Injection System Faults Diagnosis Based on Fuzzy Injection Pressure Pattern Recognition” Poceedings of the 5" World Congress on Intelligent Control and Automation, Hangzhou, P.R. China 2004 June 15-19.

# ENERGY SIMULATION

# UTILITY COMPUTING SIMULATION

Benjamin Heckmann
Ingo Stengel
Günter Turetschek
University of Applied Sciences Darmstadt
Haardtring 100
D-64295 Darmstadt, Germany
E-mail: benjamin.heckmann@gmx.de

Andy Phippen
University of Plymouth
Room 405a, Cookworthy Building, Drake Circus
Plymouth, Devon, PL4 8AA, UK

## KEYWORDS

SaaS, Cloud Computing, SOA, Service Billing, Service Provision, QoS

## ABSTRACT

Utility Computing (UC) misses an explicit definition of the core relation between IT resource utilisation, its total costs and service prices. Additionally, the implications of complex usage scenarios occurring in UC have not been examined for the service operations lifecycle. Missing those, UC service offers fail in: prediction of resource utilisation and dependent operational costs prediction, calculation of subsequent price scales, and subsequent runtime gross price calculations.

In this paper a strategy to handle UC's complexity proposing a simulation model to support each step in the service operations lifecycle is presented. The implementation approach for the model is based on OMNeT++. First simulation outcomes are presented.

## INTRODUCTION

This paper starts with a short definition of the term Utility Computing as a business model. Afterwards a common Service Operations Lifecycle is defined that has been derived from ITIL. After setting the context, the research objectives and the related research approach are introduced. Subsequent the evolved strategy that is able to handle UC's complexity is outlined. This strategy includes the demand for an UC provisioning model and a corresponding simulation model. Both models and the implementation of the simulation of the UC provisioning model are introduced. First outcomes of simulation runs are shown. Corresponding conclusions and further works are discussed.

## UTILITY COMPUTING

This work is focused on the modelling and simulation of service usage in the context of Utility Computing (UC). The term utility thereby refers to the field of industry. Here a public utility (Encyclopaedia Britannica, 2008) describes an enterprise that provides certain classes of services to a wide range of consumers.

The name Utility Computing indicates the vision of IT-based services comparable to public utilities. In this work Utility Computing is defined as a business model (Weill, 2001) for service providers offering IT-based services and charging service consumers per usage, according to (Rappa, 2004). From the provider's IT perspective UC is about service provision that is able to scale dynamically, according to real-time fluctuations in demand (Bunker *et al.*, 2006). Additionally, UC service provision offers its services equipped with the ability to charge service consumption per use (Neel, 2002).

From a consumer's perspective UC is related to "the reduction of IT-related operational costs and complexity" (Yeo *et al.*, 2006). Both perspectives, provision and consumption, have in common to target a better utilisation of generally underutilised IT resources (Andrzejak *et al.*, 2002) on both sides. In summary, UC implicitly claims an abstract description how IT resource utilisation, its total costs and service prices relate (see Figure 1).

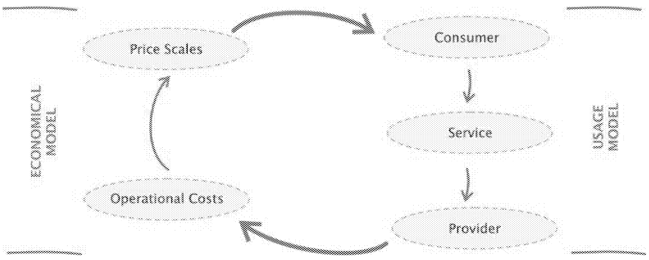


Figure 1: UC's resource – cost – price relation

Thereby Utility Computing does not refer to a specific IT service definition. From a business perspective any service that economically makes sense to be charged by its usage is addressed by UC. Therefore a more abstract service definition will be the most suitable for UC: A service represents a type of relationship-based interaction between a service provider and a service consumer to achieve a certain solution objective. (Zhang *et al.*, 2007) This definition considers the definitions of (Fitzsimmons *et al.*, 2006) from the economics perspective and (Gronroos, 2000) from the marketing perspective. From a technical perspective there are several types of services that fit into this definition, e.g. SOAP web services, HTTP web servers or Xen virtual infrastructures.

## SERVICE OPERATIONS LIFECYCLE

In the context of Utility Computing service provision a lightweight definition of a service lifecycle is necessary to obtain an overview of lifecycle stakeholders and basic activities relevant for service provision. As a basis for the definition of a lifecycle in this work, the basic lifecycle

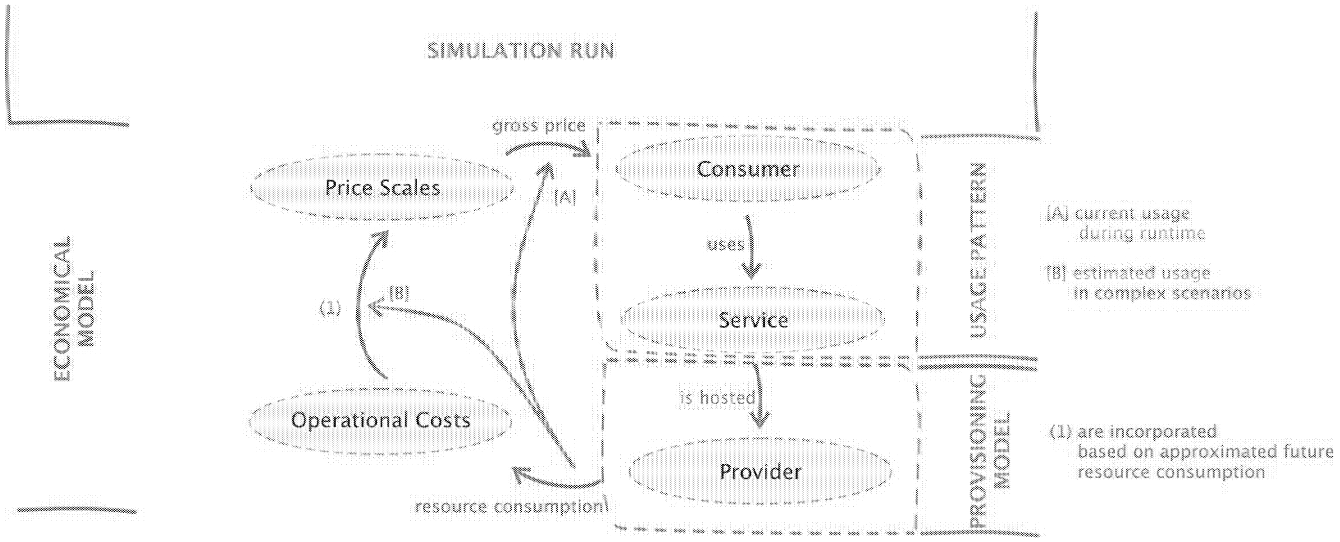


Figure 2: UC relations in the business planning

described in (Zhang et al., 2007) and the aggregation of the ITIL v3 service lifecycle described in (Beard, 2008) are used. Both descriptions can be aggregated to the three main lifecycle phases: Service business planning, service development and service operations.

The lifecycle phase of service business planning is addressing service strategy and service engagement to implement a business model. In classical IT business models, not based on the vision of UC, IT resource utilisation, its total costs and service prices only relate indirectly (see economical model in Figure 2).

During service development the service lifecycle is responsible for the design and implementation of services. This phase also includes the transition process from an implemented service to a deployed, ready for operations service. The phase service operations focuses on the provision of services. This addresses effectiveness and efficiency in delivery and support of services.

## RESEARCH OBJECTIVES

The overall context of this work focuses on specific aspects of the service operations lifecycle (SOL) for service offers based on the business model of Utility Computing. In the phase of service business planning this work refers to the corresponding service properties and service usage profiles resulting from the previous UC definition. During service development and the phase of service operations this work will focus on services in the technical context of Service-oriented Computing (SOC) (Papazoglou, 2003) consistent to the paradigm of Cloud Computing as described by (Boss *et al.*, 2007).

In this context a description of the modifications necessary to transfer a standard service operations lifecycle into a UC SOL is missing. This includes the demand for an explicit definition of UC’s core relation between IT resource utilisation, its total costs and service prices. Also specific attention must be given to the implications of complex UC usage scenarios.

The unidentified implications of complex UC usage scenarios, considerably compromise the planning, development and operation of UC service offers. Under these conditions the prediction of resource utilisation and dependent operational costs prediction, calculation of subsequent price scales, and subsequent runtime gross price calculations will fail.

## RESEARCH APPROACH

The overall work starts from the business perspective, as technical requirements depend on the business requirements imposed. Therefore, a five step approach to find solutions for the specified objectives is proposed:

(1) Describe the current state of service usage in the context of Utility Computing.

(2) Elaborate a detailed definition for the relation between a service and its consumer.

(3) Analyse the SOL of UC services.

(4) Determine the implications of complex UC usage scenarios regarding SOL.

(5) Deduct a corresponding strategy to handle the complexity.

This paper focuses on the simulation of the UC model developed as part of the overall work. The simulation, as well as the UC model, is part of the developed strategy to handle the complexity of UC usage scenarios.

## STRATEGY TO HANDLE UC’S COMPLEXITY

The overall work analyses the SOL to identify where modifications allow an optimised support for UC scenarios. Beginning with the phase of service business planning, the classical relation between resource utilisation, costs and prices is examined. Advanced relations for service provision in UC were elaborated as show in Figure 2. The relation marked with [A] adds a runtime relation between

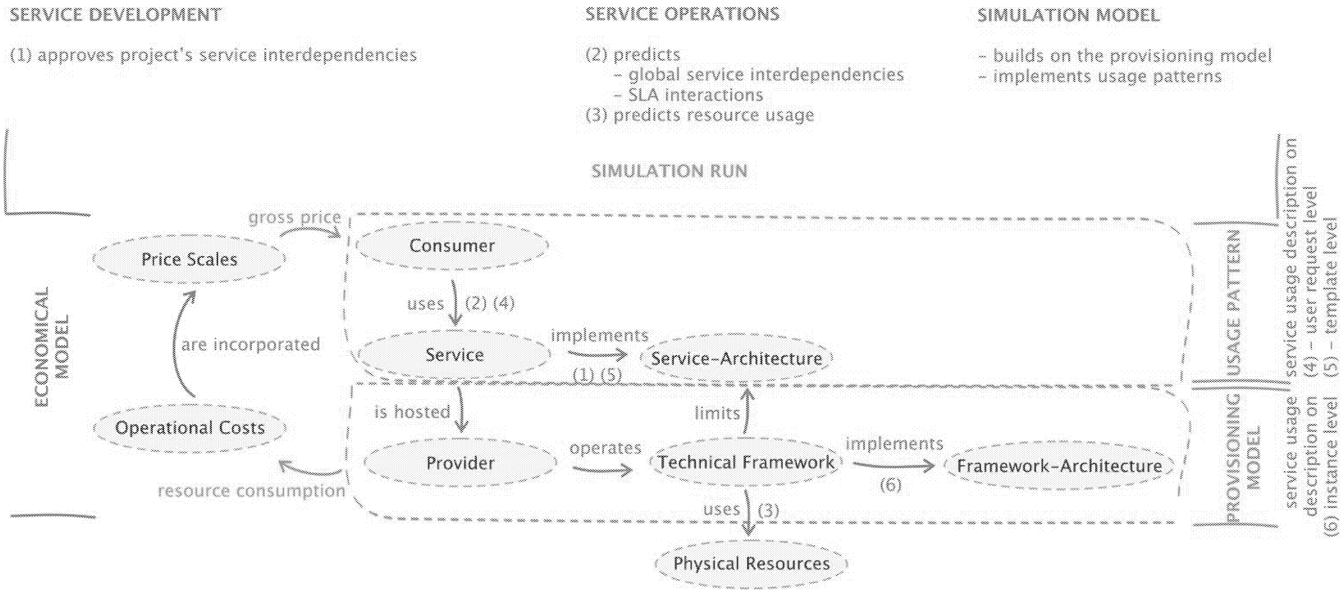


Figure 3: UC relations in service development and operations

the current usage and the gross price calculation that is essential to offer pay-per-use in UC scenarios. Beside this, service providers have to deal with complex usage scenarios, added by relation [B]. To enable the direct relation [A], constraints for UC service provision are necessary. These constraints must describe the requirements to enable this relation during service development and operation.

In the service development phase of the service lifecycle, new data including Usage Patterns specific for on-demand IT infrastructures need to be integrated into the development process (Mendoza, 2007) to improve service quality (Heckmann, 2009). To support the planning of framework architectures or the selection of framework implementations, the definition of relevant UC service provision constraints is necessary.

In the service operations phase, there are no adequate tools to evaluate service interdependencies between all hosted services. Also the Service Level Agreement (SLA) interactions between all hosted services cannot be estimated. Nor the resource planning for services to ensure contracted service levels, respecting resource consumption of other services hosted on shared resources, cannot be analysed without adequate tools for complex UC scenarios.

As a result, of the analyses of the service operations lifecycle, four major strategies for the reduction of the complexity of UC service provision can be identified:

- Define UC constraints for service architectures
- Enable the analysis of service interdependencies on development and operations level
- Permit the analysis of SLA interactions and resource prediction
- Support the proof of price scales

To implement these modifications the development of a technology-agnostic UC service provision model and a corresponding technology-abstracted UC simulation environment is proposed. See Figure 3 for a detailed overview of all previously addressed relations.

## UC MODEL

As the previous strategy suggests, the overall work defined a technology-agnostic UC model (Heckmann, 2007). In summary the model consists of eleven abstract elements, logically grouping demanded functionalities, and three basic workflows, which describe the minimum demanded interaction of those elements.

The abstract elements are consumer groups requesting services with a certain member count, request frequency and characteristics, a broker to forward requests according to costs aspects, a load-balancer to forward respecting load aspects of a request, a host offering resources such as computing cycles, memory, storage and network, and service instances consuming offered resources. Additionally some elements to organise service provision: a registry, monitoring, and a service type element. In a derived technical IT architecture these functional groups can be represented as standalone components, but could also be combined in joint architectural elements. As basic workflows a simple service consumption workflow, a complex service consumption workflow, and a cascaded service consumption workflow where defined.

Other models in this context have been proposed by (Mendoza, 2007), (Zhang *et al.*, 2007) or (Bunker et al., 2006). The model of Bunker and Thomson is the most inadequate of them, since it provides too few details to be helpful for IT architects to design a suitable UC architecture for a specific service provision scenario. The model delivers only a quick overall IT strategy to the provision of UC-based services. The UC model by Zhang, Zhang and Cai was developed from a business management perspective. It specifically aims to the provision of SOAP-based web services and describes in detail how those should be provided. While the model of Mendoza uses a very efficient model building approach, starting from a

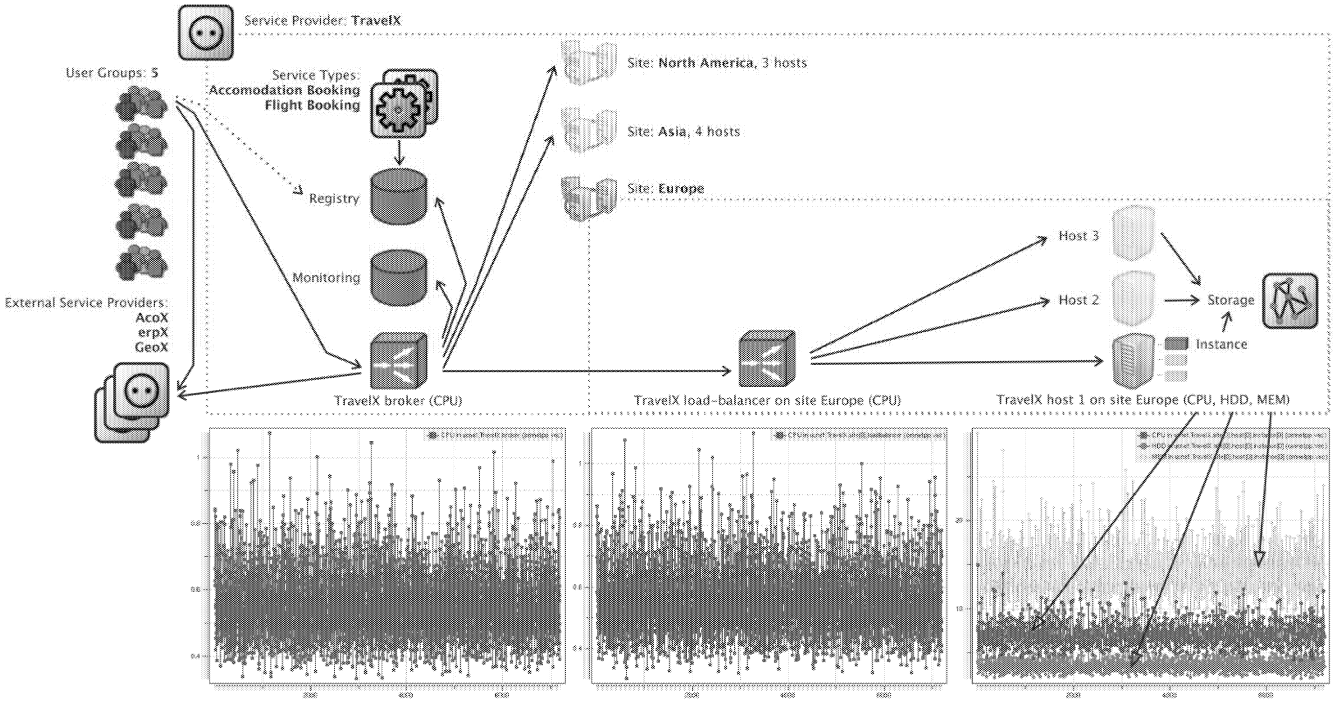


Figure 4: Simulation model as multi-tier architecture

technological perspective. Both of the afore mentioned models are very complex and technology-dependent. Therefore a custom model building was conducted, targeting a lightweight technology-agnostic solution.

## SIMULATION MODEL

The simulation model represents a multi-tier architecture (see Figure 4) for the UC-conform provision of services in service-oriented computing. The functionalities described in the UC model have been transformed into the simulation model that implements this architecture. The current implementation is capable to simulate:

- Complex user behaviour (user group): Messages can be sent with random or fixed timeslots to control the amount of messages arriving at the broker. The resource consumption for transport and processing of the embedded service request can be determined separately. It is also possible to configure transport priorities. Each service request can include a free number of subrequests to represent service cascades.

- Resource measurement and monitoring for computing cycles, memory and disk space: The hardware resources simulated and monitored are computing cycles, memory and disk space. Network traffic (bandwidth and delay) is currently not monitored, but simulated. Additionally monitored are the load-balancer and broker queues and their message transport resource consumption. The overall resource consumption for the storage network is also traced.

- Message billing to service consumers (broker): To each request response a bill based on the processing sites computing cycles, memory and storage costs will be attached. The consumption of these resources during processing of the request is billed, and it is possible to add additional per site and per consumer margins.

- Message routing by site costs (broker): Messages get routed to a site with enough resources to process the request and the least costs for processing.

- Message routing by resource demand (load-balancer): Messages are routed by a site's load-balancer to a host with enough resources.

- Message queuing (broker & load-balancer): Messages are temporarily stored within the service broker or service load-balancer when not enough resources for their processing are available. They are recalled from queue after a certain scheduling time and entered again in the scheduling sequence of either the service broker or the service load-balancer. In doing so the queuing consumes resources in the system, and if the system balancer runs out of resources, incoming messages are dropped.

The simulation model is implemented based on the discrete event simulation environment OMNeT++ (Varga, 2001). For each simulation run it is possible to determine the number of user groups requesting services. It is possible to vary the total number of group members, the behaviour timing as well as the type of request in meanings of service type and request complexity individually per user group. It is possible to specify any service type and any number of service providers. Each provider may have several sites with any number of hosts. Each host can be individually equipped with computing power, memory and storage. Additionally each site has access to a storage network to estimate storage network loads. For each user group and service provider the price relation to each service type can be individually adapted. This highly flexible configuration targets the necessity to represent complex UC scenarios.

## FIRST OUTCOMES

The largest test scenario currently simulated represents a virtual travel booking processor with 2770 consumers in five groups, each sending a single request of the same service type including two subrequests to external service providers. Thereby each request's initialisation is randomly scheduled within a given timeframe. As outcome of each simulation run a data pool consisting of all values stated in the resource measurement and monitoring definition of the simulation model is provided. As examples for graphs based on parts of the outcomes, in Figure 4 the consumption of computing cycles characterised as the CPU utilisation is shown for the TravelX broker, a load-balancer and a host. The host's graph also shows the memory (MEM) and disk space (HDD) utilisation on the contemplated host.

Additionally to simulation runs testing a large amount of concurrent users, the implementation showed that it is able to simulate the behaviour of highly meshed service cascades. Even if it comes to special cases like looping service requests, where subrequest providers themselves use services provided by the original subrequest initiator.

Or in case of internal subrequests occurring, where the provision of a service involves requests to other internally provided services.

## CONCLUSIONS AND FURTHER WORK

This paper identifies the modifications necessary to transfer the standard SOL into a UC SOL. As part of the presented strategy to handle UC's complexity a simulation model is introduced. First tests of this simulation model have shown that it is possible to represent complex scenarios. The prediction of resource utilisation, dependent operational costs and subsequent runtime gross prices has been shown in virtual scenarios. Further the simulation model mustbe validated analysing real world scenarios. The main aspect for adequate representation of the service's behaviour will be the calibration of the simulation runs to reflect the current resource consumption of service requests. Here further research has to be conducted.

Also part of future research must be the documentation of theoretical aspects of the simulation model building. This includes the relation between discrete event simulation and queuing concepts from queuing theory, the revision of relevant probability topics and the relevant background in stochastic processes.

It is assumed that the technology-abstracted simulation model can also be used for the simulation of RESTful or simple services, such as web servers. Here further research will be conducted.

## BIOGRAPHIES

Benjamin Heckmann is a researcher at the aiDa research center in Dieburg and PhD student at the University of Plymouth, UK. He holds a M.Sc. in computer science. His research interests are in the areas of Utility Computing, Cloud Computing, Unified Communications and IT-Security.

Ingo Stengel graduated at the Cork Institute of Technology, Ireland. He is co-founder and Executive Director of the igdv-Centre for Advanced Learning, Media and Simulation at the University of Applied Sciences Darmstadt. His research interests are in the area of Multiagent-Systems, Simulation Software, IT-Security and Advanced Learning.

Andy Phippen received his PhD in the year 2001. He is Senior Lecturer in Business Enterprise and Ethics at the University of Plymouth. His research focuses on the impact of software development and learning & teaching in higher education. Further, he is the director of the IT liaison at the University of Plymouth.

Günter Turetschek is Professor for computer science at the University of Applied Sciences Darmstadt; co-founder and director of the Institute for Applied Informatics Darmstadt (aiDa). His research interests are in the area of Business Computing, Unified Communications and Utility Computing.

## REFERENCES

Andrzejak, A., J. Rolia and M. Arlitt. 2002. "Bounding Resource Savings of Utility Computing Models". HP Labs Technical Report HPL-2002-339.

Beard, H. 2008. *Cloud Computing Best Practices for Managing and Measuring Processes for On-Demand Computing, Applications and Data Centers in the Cloud with Slas*. Emereo Pty Ltd.

Boss, G., P. Malladi, D. Quan, L. Legregni and H. Hall. 2007. "Cloud computing". IBM, developerWorks, WebSphere, High Performance On Demand Solutions.

Bunker, G.; and D. Thompson. 2006. *Delivering Utility Computing: Business-driven IT Optimization*. John Wiley & Sons.

Encyclopaedia Britannica, 2008. *public utility*. In Encyclopaedia Britannica Online, retrieved December, 2008.

Fitzsimmons, J.A.; and M.J. Fitzsimmons. 2006. *Service Management: Operations, Strategy, and Information Technology*. 5th Ed., Irwin/McGraw-Hill, Homewood, IL.

Gronroos, C. 2000. *Service Management and Marketing: A Customer Relationship Management Approach*. John Wiley & Sons.

Heckmann, B. 2007. "Service provision in a utility computing environment". SEIN 2007, University of Plymouth, 14-15 June 2007.

Heckmann, B. 2009. "Technology-agnostic definition of the Utility Computing service operations lifecycle". Transfer Report, University of Plymouth, April 2009.

Mendoza, A., 2007. *Utility Computing Technologies, Standards, and Strategies*. Artech House Inc.

Neel, D. 2002. "The Utility Computing Promise". InfoWorld, April 12, 2002.

Papazoglou, M.P. 2003. "Service-oriented computing: concepts, characteristics and directions". Web Information Systems

Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on 10-12 Dec. 2003 Page(s):3 – 12.

Rappa, M.A. 2004. “The utility business model and the future of computing services”. IBM Syst. J. 43, 1 (Jan. 2004), 32-42.

Yeo, C.S., M.D. Assunção, J. Yu, A. Sulistio, S. Venugopal, M. Placek and R. Buyya. 2006. “Utility Computing on Global Grids”. Hossein Bidgoli (ed), The Handbook of Computer Networks, John Wiley & Sons, New York, USA, accepted in April 2006 and in print.

Varga, A. 1997. “Flexible topology description language for simulation programs”. Simulation in industry: 9th European Simulation Symposium 1997:225-229.

Varga, A. 2001. “The OMNeT++ Discrete Event Simulation System”. In the Proceedings of the European Simulation Multiconference (ESM'2001). June 6-9, 2001. Prague, Czech Republic.

Weill, P. and M.R. Vitale. 2001. “Place to space: Migrating to eBusiness Models”. Boston, Harvard Business School Press.

Zhang, L.-J.; J. Zhang; and H. Cai. 2007. *Services Computing, Core Enabling Technology of the Modern Services Industry*. published by Springer and Tsinghua University Press.

# THE SIMULATION OF THE ENERGETIC CHARACTERISTICS OF HYDROPOWER PLANTS AND WIND FARMS APPLIED TO THE COST ANALYSIS OF ELECTRICITY GENERATION IN THE POWER SYSTEM

Eugeniusz M. Sroczan
Institute of Electric Power Engineering
Poznań University of Technology
ul. Piotrowo 3A 60-965 Poznan, Poland
E-mail: eugeniusz.sroczan@put.poznan.pl

**KEYWORDS**
Computer aided analysis, energy management, decision support system, interactive simulation, optimization.

## ABSTRACT

The cost of electricity generation, in the given power system (PS) covering the demanded load during the defined time horizon, depends on the structure of committing power plants connected to the power network. The simulated costs of electricity are calculated by using various methods depending on primary energy used in the concerned power plants. The results of simulation help to make decisions concerning the strategy of the power plant operation in order to meet the PS balance – between demanded load $P_{dt}$ and generated power $P_{gt}$ , in each *t* moment of time.

In the typical local PS one can encounter the storage hydropower plants (SHPP), pumped-storage hydroelectric plant (PSHP) and wind power plants (WPP) committing as well as standard thermal power plants (TPP) and combined heat and power plants (C-HPP).

The applied procedures simulate the costs of the generation and verify from the economic point of view the quality of balance of the generated and demanded electric power and the energy with regard to the input-output characteristics of considered kinds of power units. These characteristics depend on variable fuel quality, weather conditions, technical state of power units and the power grid constraints.

## INTRODUCTION

Usually, the applied simulators enable the optimization of the electric energy cost by using some procedures that fix the load of sources committed in the discussed PS (Sroczan 2008; Sroczan 2005, Baltierra at al. 1998) The essential problem of the local energy market (LEM) is to balance the energy production with consumers' demands, with regard to the technical and economic boundaries, as well as legal restrictions outlined by EC law in the area of the environment preservation.

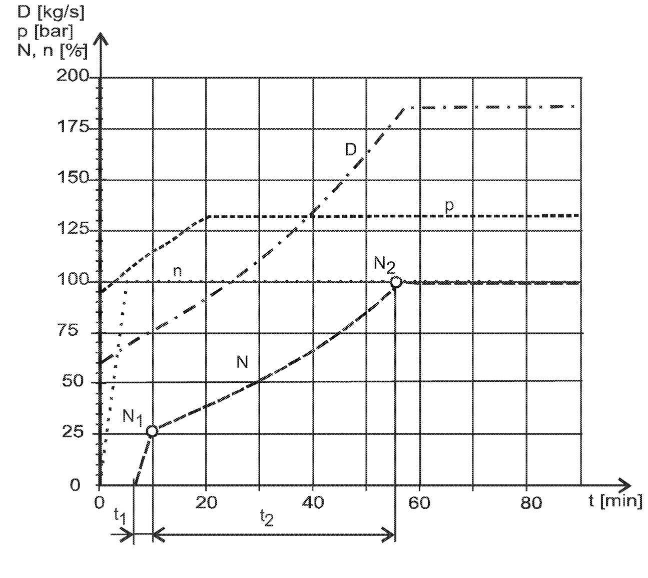
The structure of the developed simulator consists of procedures applying the classical attempt to optimization as well as the fuzzy technique. The obtained characteristic are prepared with the use of SCADA system and after preprocessing are applied to simulation of the dynamic properties of concerned power plants, especially some possibilities of change the covered load in the mode of cut-in and cut-off or shut down.

## TECHNICAL CONSTRAINST OF ELECTRICITY GENERATION

In the group of technical requirements, the possible load changeability of the plant - the proper gradient of covered load - has been taken into account as it affects the cost of generation caused by the time of uneconomic operation. The time duration of full load of thermal unit is much greater in comparison with the SHPP or WPP (Fig. 1). For simulation purposes this time is calculated as $t_{FL}$:

$$t_{FL} = t_1 + t_2 = \frac{N_1}{k_1} + \frac{100}{k_2} \ln \frac{N_2}{N_1} \text{ [min]} \qquad (1)$$

where: $N_1$ [MW]– maximal value of linear load increase, $k_1$ [MW min$^{-1}$]– linear increase of power, $N_2$ , $N_1$ [MW] – rated and momentary power of the unit, $k_2$ [%] – coefficient of constant percentage increase of load of power unit.



Figures 1: Path of Optimal Gradient of Steam Turbine Parameters in the Mode of Increasing the Load *N* of Thermal Power Unit

In case where the power of the given source depends on wind speed or water inflow, the time schedule program of load must take into consideration some uncertainty of disposed volume of power or energy. The lack of power in that source node is compensated by additional flow from neighboring nodes of power grid or reserve power plant. This kind of possibilities depends on the structure of generation and allocation of power system reserve and flexibility of power grid.

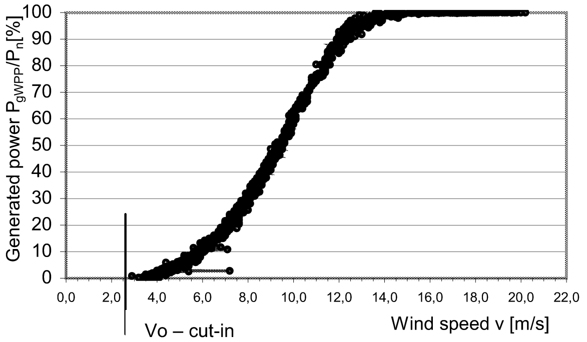
**Wind power plant**

Theoretical value of power of wind turbine measured on the shaft is calculated as:

$$P_{WTt} = \frac{1}{2} C_p \rho \cdot \pi \frac{D^2}{4} v_t^{\ 3} \cdot 10^{-3} \quad \text{[kW]} \qquad (2)$$

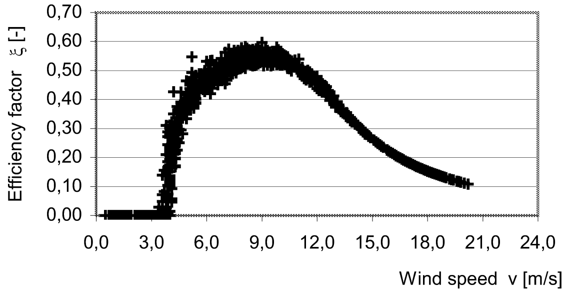
where: Cp [-] – efficiency coefficient, $\rho_o$ – air mass [kg/m$^3$], D [m] – rotor diameter, $v_t$ [m/s] – momentary wind speed.

The value of generated power depends on input-output characteristic of wind turbine (fig. 2.) and value of efficiency coefficient depending on wind speed and rotor parameters (fig. 3.).



Figures 2: Input-output Characteristic of Wind Power Plant

Varied wind speed is affecting the change of energetic characteristic of wind farm and therefore the participation of wind power plant in load dispatch is modified.



Figures 3: Efficiency Factor ξ(v) of Wind Power Plant, Calculated as a Function of Wind Speed

During the generation time the coefficient ξ (fig. 3.) is monitored to check the proper operation of the wind plant.

**Thermal plant**

Energetic characteristics of thermal plant *Q(P$_g$)* are updated by using the correction coefficients for crucial parameters, which affects the efficiency factor of the power unit. For load dispatch purposes the corrected and guaranteed characteristics of heat consumption *Q(P$_g$)* are calculated as:

$$Q(P) = a \cdot P^2 + bP + c \quad \text{[GJ/h]} \qquad (3)$$

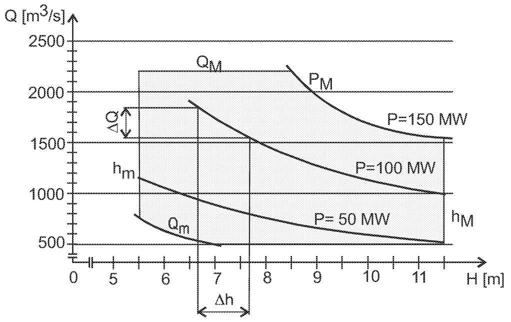
where: a, b, c – coefficients, P [MW] – actual value of power unit load.

**Hydropower plant**

The value of power generated in hydropower plant is defined as:

$$P_{HPP} = 9{,}81 \cdot Q_t \cdot H_t \cdot \eta(Q_t, H_t) \quad \text{[kW]} \qquad (4)$$

where: Q [m$^3$/s] – water flow, H [m] – head of water, *η(Q,H)* – turbine efficiency coefficient.



Figures 4: Characteristics of Water Consumption in the Hydropower Plant

The load of power grid causes additional losses of power and energy in the power line, which causes a decrease in the efficiency of conversion of the primary energy into the electric one. Therefore the structure of generation affects the economy of generation due to elasticity of transmission grids.

## CALCULATING THE CHARACTERISTICS FOR POWER LOAD DISPATCH

The main aim of this paper is to simulate the effect of the structure of the electric power sources subset and flexibility of power plants in order to minimize the costs of energy with respect to renewable energy ratio (Sroczan 2005) and operation of committing power plants with the use of updated characteristics.

The proposed attempt is based on the algorithm considering the real costs of energy in the given circumstances calculated with regard to demand and the updated characteristics of loading the renewable energy sources SHPP (4) and WPP (2). The costs are calculated for hydro and wind power plants as well as for the thermal plants fired with hard and brown coal (3).

The decision of load for a plant is optimal if *C$_{PSt}$* - the costs of generation in the PS will fulfill, in each time *t,* the following relationship (Sroczan 2005):

$$C_t = \min\left\{\sum_{i=1}^{n} C_i\left(P_{gi}\right)\right\} \qquad (5)$$

where: P$_{gi}$ – the level of generated power in i-th committing power plant.

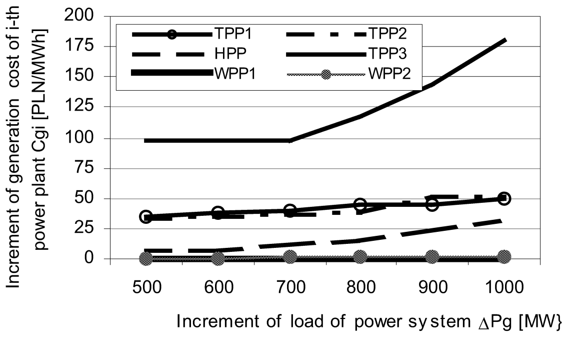
The process of optimisation of the work of committing power plants considers time period $T$ in which all of the discussed units change the value of generated power with accordance to power demanded by end users. The goal is defined as minimization of generation cost (5) with respect to the PS constraints.

The relationship between the increasing cost of generation by thermal plants and coefficients: $\gamma$ [PLN/m$^3$]- cost of stored water in SHPP and $\kappa$ [PLN] – equivalent cost of wind (Sroczan 2005) affects the structure and the value of the load of the subset of HPP and WPP. The optimal case for renewable sources occurs when:

$\gamma \frac{\partial W}{\partial P} = \kappa \frac{\partial C}{\partial P} = \min \left\{ \frac{\partial C}{\partial P} \right\}$ and both types of renewable sources of energy – sets of HPP and WPP are loaded (fig. 5). The “green energy” is converted in the desired volume, defined by PS operator, using the RER coefficient (Sroczan 2008a). Finally the results define the policy for operating the set of power plants of the given PS, with regard to data processed by SCADA systems (Sroczan 2008b). The costs of power and energy containing varied and fixed costs of generation are increasing due to the internalization of outside costs.

Energy costs include fuel burned – delivered cost of fuel (base cost, escalations, premiums or/and penalties, transportation, demurrage), outside laboratories fees and other outside costs related to the fuel procurement.

The mentioned above attributes of power plants are considered in the process of optimization of the cost of generated electric energy.



Figures 5: Updated Characteristics of Load of Committing Power Plants

In the presented attempt to analysis of $C_{PSt}$ the RER (renewable energy ratio) value takes into account the disposed volume of renewable sources in discussed PS (EU regulations) and at the same time enables the possibilities of changing the structure of generation.

More over the time of full load (1) is taken into consideration for the thermal units (fig. 1).

The analysis of costs of generated energy, calculated with the use of developed simulator allows to draw the following conclusion. The participation of power plants supplied with renewable energy sources causes the increase of the cost of generation in the discussed PS, due to additional costs stated by the rules of energy managing and these costs are transferred to the end users of energy.

## CONCLUSIONS

The developed procedures enable calculation of the expected costs of energy for the given set of committing power plants, assuming the varied structure of primary energy sources and types of power plants. The effect of PS sources structure on generation costs of electric energy is calculated using corrected energetic characteristics of plants and developed procedures of simulation. These procedures allow for the modeling of the optimal policy of operation of the power plant with regard to the PS constraints and regulations, introduced by EU directives, that specify the volume of energy required to be obtained from the renewable sources.

The developed simulator generates a range of scenarios of the load of committing units with regard to the dynamic properties of considered power plants and varied in the time resources of wind and water energy.

## REFERENCES

Baltierra A.E.; Moitre D.; Hernandez J.L.; Aromataris L. 1998. “Simulation of an Optimal Economic Strategy of a Wholesale Competitive Electric Energy Market”. In *Proc. of 10$^{th}$ European Simulation Symposium*. Nottingham, England 1998. 255-259.

Jasiński P., Kaproń H., Optymalizacja pracy elektrowni w warunkach ograniczonej konkurencji. *Rynek Energii* nr 2(75)/2008 r. p. 24-30. ISSN 1425-5960 (in polish).

Malko J., Wilczyński A. 2009. Electricity markets and regulation of CIGRE-Study Committee C5. *Rynek Energii* 2(81). p. 23 – 31. ISSN 1425-5960 (in polish).

Sroczan E. 2008a. “The Simulation of the Procedure for Multiattribute Choice of the Structure of the Power Plants in the *European Simulation and Modelling Conference 2008*. Ed.: Bertelle C., Fortino G. Publ. EUROSIS-ETI. Ghent Belgium. p. 521-523.

Sroczan E.M. 2008b. Cost optimization of the electric power supply by using the IT system. *Rynek Energii* 1(74). p. 18-22. ISSN 1425-5960 (in polish).

Sroczan E. 2005. “The Simulation of Power System Structure Effects on Technical and Economic Effecttiveness of Energy Generation.” In *The 2005 Simulation and Modelling Conference*. University of Porto Portugal. 24-26 Oct. 2005, 391-395.

Wind turbines farm – chosen exploitation data 2008-2009. EEZ Ltd. Warszawa.

**EUGENIUSZ SROCZAN** is employed as an assistant professor at the Poznan University of Technology (PUT) and professor of State Higher Vocational School in Gniezno. He obtained from PUT a M.Sc., and Engineering Degree in area of Industry Automatic and a Ph.D. in area of Electric Power System Engineering from PUT. Author and co-author of papers on Power System Economic Operation, Energy Management Systems in Industry and Automation of Energetic Processes as well as Water and Waste-Water Treatment Plant.

Author of the book on contemporary electrical installations of home. Since 2005 he has been the head of The Institute of Computing Science in State Higher Vocational School in Gniezno and since 1984 he has been the President of Branch of Polish Electriciants Society at the PUT.

# PARALLELISM OF CONTROL PROCESS OF ELECTRIC ENERGY CONSUMPTION IN A DISTRIBUTED POWER GRID SYSTEM

Dariusz Bober and Henryk Kapron
Faculty of Electrical Engineering and Computer Science
Lublin University of Technology
Nadbystrzycka 36, 20-618 Lublin,
Poland
E-mail: {d.bober, h.kapron}@pollub.pl

**KEYWORDS**
Power modes model, DMS, smart grids.

## ABSTRACT

The abilities of simulation applications give the powerful tools for the scientists and enginiers. It is obvious that before implementation of some idea you ought to test it in simulation environment. In the article authors present the idea of new model of electrical energy consumer powering – the power modes model. The model is implemented in Matlab® environment and it has been tested for the real parameters and data of one Polish power supplier of 2007 rear spectrum. The proposed methods of data metering and acquisitions allows for the parallelity of control process of electric energy consumption in a distributed environment of a power grid system, or a part of that system.

## INTRODUCTION

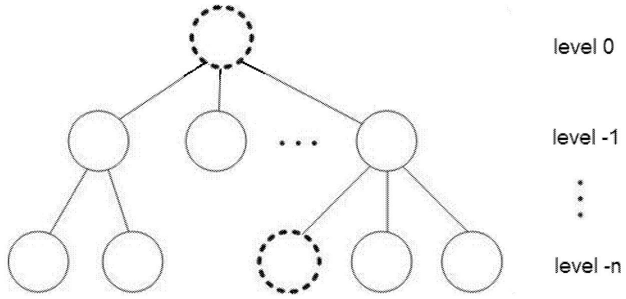
In the most, a power grid system could be presented as a hierarchical structure (fig. 1). Where the node on the top is a supplier, which distribute the power to the lower level of hierarchy. The nodes of that level represent e.g. the departments of the supplier, the power stations an substations, the sub areas of the whole geographical area where the supplier distribute the energy. At the lowest level of the hierarchy are the energy consumers. More:

- one node of energy consumer could represent an internal hierarchy of energy distribution grid, with similar structure to fig. 1, where the main node represents the consumer and the sub nodes are his production objects and the electrical equipments on the lowest level.
- and on the other side the whole structure of the supplier (fig. 1) could be an subsystem of the global power system, and then the main node of the hierarchy (fig. 1) become an node on the lowest level of the global hierarchy – the transparency of mark line-node functionality.

It is important that on each node of each level of the hierarchy the true is (Gladys and Malta 1999):

$$P^{S}_{(\text{level } 0)} = f(x, \Sigma P^{D}_{(\text{level } -1)}). \qquad (1)$$

The power supply $P^S$ ability (1) of a node on level 0 is a function f of sub nodes power demand summary and is also determinate by a technical condition x of power grid infrastructure. On the other hand the power demand $P^D$ of one node of level -1 depends of the power supply of its supplier and its neighbours – the others nodes of the level -1. So the main node is interest to order/generate such enough power such his sub nodes demand. On the other side the sub nodes of the node couldn't demand more power that the node is able to supply. In this situation it is necessary to introduce some power supply limitation. The sub nodes ought to calculate the risk of the lack of electricity situations.



Figures 1: The hierarchical structure of electric energy distribution process – power system. The marked line-nodes presents the transparency of node functions – a node of the lowest lever - a consumer - becomes the supplier in its internal distribution grid.

The cost of lack of electricity (Gabrysiak 2004; Paska 2005) and the newest blackouts history (Malko 2006) necessitate the need for change and a new better solutions developing. The power modes model - presented in the next chapter – is helpful for the solution researching.

## REMODELING OF THE ELECTRIC ENERGY CONSUMPTION STRUCTURE

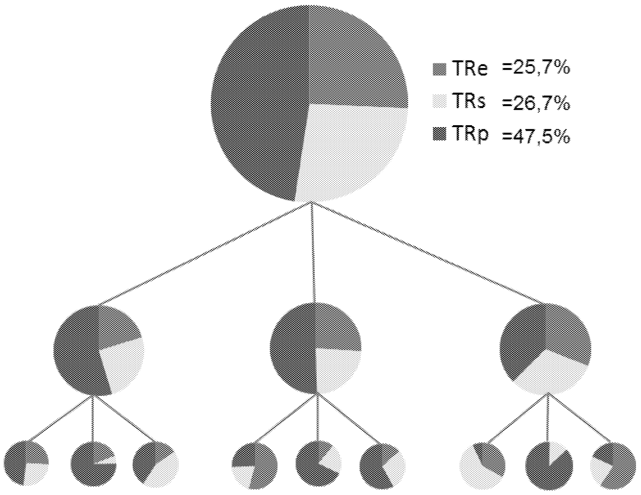
The researches (Bober 2008a) of the households' consumers' preferences and decisions of energy consumption limitation in the situations of power deficit allows for introduce new model of electricity consumer powering "the power modes model" (Bober 2008b), where "a part" of consumed energy E is associated with a quality parameter q:

$$TR = g(E, q). \qquad (2)$$

The quality parameter q described some individual principles of each power mode TR and the conditions (eq. energy price, hours of access, degree of reliability, etc.) of energy consumption by the stuff powered in the power mode. So, the described households could be powered by three power modes: protected power mode TRp, standard power mode TRs and economical power mode TRe. The energy consumed by the households in the new model (2) will be the sum of the modes:

$$E = E_{TRp} + E_{TRs} + E_{TRe}. \qquad (3)$$

The model of power modes significantly simplifies the process of energy consumption control. There is no necessity to detail control of each node of each level of the power system grid (see, fig. 1). Each node of the grid could process the control itself and the same it could manages the energy consumption structure in this part of hierarchy where it is the main node. The idea of distributed control of energy consumption structure in a hierarchy of a power grid system is presented at fig. 2.

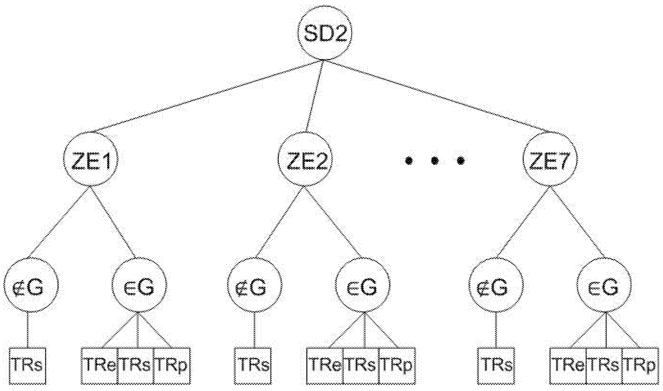


Figures 2: The structure of a power system hierarchy with the power modes model implementation (Bober 2008c).

Although the presented idea looks promising, but there is a lot of conditions to be fulfilled before the power modes model will be introduced into practice. Some aspects of the solution implementation were described in (Bober and Kapron 2009). In this paper, we concentrate on simulations of possibilities of the power modes model in the implemented in the Matlab environment.

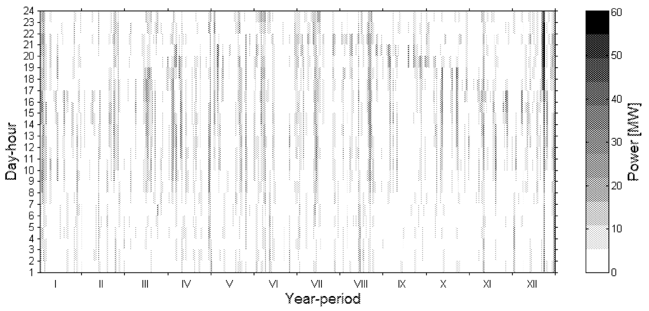
## SIMULATIONS OF THE POWER MODES MODEL POSSISBILITIES

We have implemented one of the polish electricity distributor structures (see, fig. 3). The distributor (with code name SD2) has divided his administrative area into seven divisions' sub-areas, with symbolic name ZEx. He distributes energy to many types of consumers, but we divide them into two groups: households, which buy the energy in tariff "G" and the other which do not belongs to tariff "G". For group of consumers who $\notin$G we have linked the standard power mode TRs.



Figures 3: The structure of controlled object (Bober 2008a).

For this structure of the controlled object we have received the real data of day-hour power demand of 2007 year from the distributor. We decide that the distributor hypothetical day-hour power supply for this year will be his data of day-hour demand prognoses. The received data we have imputed into the Matlab® simulation environment as 24hours x 365days matrixes and we minus them to see the research area (see fig. 4). The problem to be resolved is the power demand decreasing to reduce the dark areas on deficit matrix.



Figures 4: The power "deficit" matrix. The dark areas points the hours where the customers'' power demand overloads the distributor's demand prognosis.

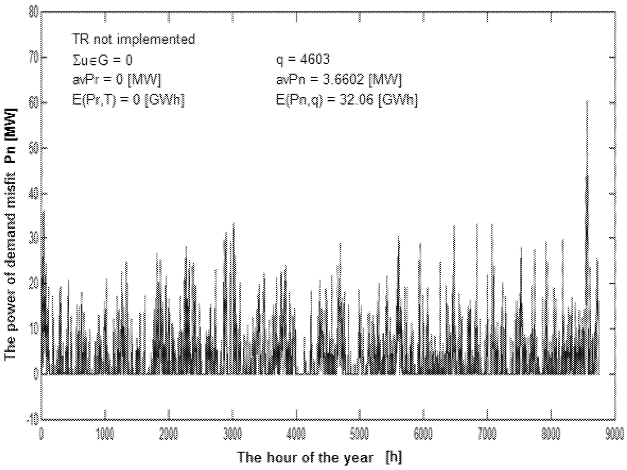
We will research how the knowledge about the power modes' structure of the households' customers will help us to resolve the problem. We try to find out how the volume of power mode TRe of energy consumption returned to the power system will help us in the power "deficit" compensation? It is interest, because the "deficit" is generated by the whole distributors' clients, not only by the households. If by the power modes model we control the energy consumption of households and in this way we significantly change the power demand of the whole distributors'' consumers – it will be a very good result. And as it is presented in the next subchapter this target has been achieved.

We define the indicators of the object state measurement:

- TR – id's of ZE nodes where the TR model was implemented;
- $\Sigma u \in G$ – numbers of households where the TR model is implemented;
- avPr – the average power demanded by households powered in the TR model;

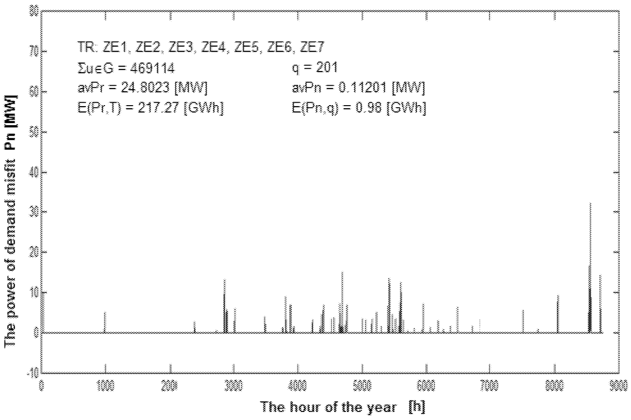
- E(Pr,T) – the total energy consumed by households powered in the TR model in the period of the whole year;
- q – numbers of situations where the supplied power do not balance the demand of the consumers, in this situation there is necessity to buy some more energy/power from the generators;
- avPn – the average power “deficit”;
- E(Pn,q) – the total energy “deficit”.

The initial object state is presented at fig. 5.



Figures 5: The state of the object before power modes model implementing, the whole nodes/consumers consume the energy without restrictions. The picks of the graph corresponds with dark areas of fig. 4.

In the consequence of series simulations of the power modes control of the object hierarchy (see, fig 5) we received the state of the object as is presented at fig. 6.



Figures 8: The final state of the object.

As you can see, the indicator E(Pn,q) of the whole year energy consumed over the prognosis demand decreases from 32,06 GWh at the state before any control simulation (see, fig. 7), to less than 1 GWh after the restriction of TRe mode simulated on selected division-nodes. The other state indicators also look better.

The details of the experiment are described in (Bober 2008a).

## CONCLUSIONS

There is common interest of the power systems improvement (Billewicz 2007, Kapron 2007, Malko 2009, Sroczan 2007) especially in the aspects of the system reliability and electricity sufficiency. The presented results of simulation of the power modes model implementation in the condition of real distributor shows that the model is helpful for the electric energy control. By distribution of decision-making process of control into dispersed “smart” nodes, the process of control could be parallelized into numerous independent processes. In consequence, that will increase the object control process performance.

## REFERENCES

Billewicz K. 2007. “System of automatic meter reading of electricity AMR”. In *Measurement Automation Robotics* 7-8/2007, 9-12.

Bober D. 2008a. *Hierarchical Control System of Electricity Consumption*. PhD dissertation. Lublin University of Technology.

Bober D. 2008b. “The electric energy customer powering by power modes”. In *Energy Market*, No 1 (Feb), 27-32.

Bober D. and H. Kapron. 2009. “Distributed system for data acquisition and management of electric energy consumption”. In *IEEE International Workshop on IDAACS’ 2009* Rende (Coseza), Italy, in pending.

Gabrysiak A. 2004. “One day of lack of electricity”, In *Around the energy*. Termedia. (Apr), 17-23.

Gladys H. and R. Malta. 1999. *The generator works in a power system*. WNT. Warsaw.

Kapron H. 2007.”’Energy market, producers’ competition”. In *Energy Market*, No 6 (Jun), 13-16.

Malko J. 2006. Blackouts: the effect of bad management and politics. In *Around the energy*. Termedia. (Feb), 17-23.

Malko J. 2009. “Intelligent networks - principles and technologies”. In *Energy Market,* No 3 (Mar), 9-17.

Mesarovic M. D and Y. Takahara. 1978. *General systems theory: Mathematical foundations*. Mir. Moscow.

Moisiejew N. N. 1983. *Elements of optimal systems theory*. WNT. Warsaw.

Paska J. 2005. *Reliability of power systems*. Warsaw University of Technology. Warsaw.

Sroczan E. 2007. “Application of IT system to optimize the cost of the electricity supply”. In *Energy Market*, no 1 (Feb), 18-22.

## BIOGRAPHY

**HENRYK KAPRON** Professor at the Lublin University of Technology, Faculty of Electrical Engineering and Computer Science. Chief of Power Generation and Economy Department. From 1995, prof. Kapron has been come to the CHP Lublin-Wrotkow Ltd. as a member of control exploitation team. He is also Editor-in Chief of Energy Market Journal.

**DARIUSZ BOBER** assistant at the Lublin University of Technology, Faculty of Electrical Engineering and Computer Science. Computer network administrator at Lubella. He specializes in the databases, the web services and the XML technology. He is also interested in the power economy and the systems of control.

# HEALTH SERVICE MANAGEMENT

# Human resources integration in complex systems modeling: application to the health care systems

Michelle Chabrol
LIMOS CNRS UMR 6158
Blaise Pascal University
63173 Aubière, France
(+33)4 73 40 50 34
chabrol@isima.fr

Michel Gourgand
LIMOS CNRS UMR 6158
Blaise Pascal University
63173 Aubière, France
(+33)4 73 40 75 14
gourgand@isima.fr

Sophie Rodier
LIMOS CNRS UMR 6158
& University Hospital of Clermont-Fd
63058 Clermont-Fd Cedex 1, France
(+33)4 73 75 04 37
rodier@isima.fr

**KEYWORDS**
Specification language, modeling, simulation, healthcare systems.

## ABSTRACT

The human factor represents a key element of the enterprise. In a constantly changing environment, it ensures timeless and competitiveness. In this paper, we are interested in complex systems modeling and, in particular, in the human resources integration. We propose a methodology and an analysis language in order to formalize the knowledge and to design decision-making tools which cover all the temporal horizons (strategic, tactic and operational). We present the language characteristics and an application in the health care systems.

## 1. INTRODUCTION

In this paper, we are interested in discrete systems composed of resources whose main objective is the manufacturing or processing of goods or services. These systems can present a structural and functional complexity also called systemic complexity which makes difficult the evaluation of the performance criteria.

Modeling is thus an essential stage to complete for the understanding of such systems, prior to the design of decision-making tools which are adapted to the domain.

One of the difficulties we face in such systems is the human resources integration that is to say the human resources consideration in the process modeling. The human factor represents a key element of the enterprise. In a constantly changing environment, it ensures timeless and competitiveness. It also represents an important part of the costs of the structures. So, it is now necessary to model and analyze the enterprise processes by taking into account the human resources as well as material resources.

We consider the health care systems as complex systems. We remind a few of these systems specificities in [5]. The main complexity results in a significant quantity of patient pathways, as well as in the complexity of most of these pathways which are often not linear. It induces as well the use of probabilities and management rules in these pathways, and complex operations on human resources, dealing with the resources assignments and priority rules, along with their associations with other human resources used for the same operation. These associations result in the combinations of several human resources for the same patient, and for the same operation.

In this paper, we are interested in the modeling of such systems and, in particular, in the human resources integration. These works take place in the modeling project of a new hospital. Our main objective is to provide to the hospital managers a set of decision making tools based on operation research methods and tools such as simulation models, optimization methods…

After the study of the existing tools, we propose a methodology and a language of analysis and evaluation of the systems (LAES) to take into account resources combinations specificities. We propose then the application of this language to the health care systems.

## 2. STATE OF THE ART

Modeling languages derive from different scientific communities. In [6], the authors note that since the first development in the area of enterprise modeling started in the US in 70s, e.g. Structured Analysis and Design Technique (SADT), Integration Definition for Function Modeling (IDEFx, http://www.idef.com/)[13], Data Flow Diagram…, a lot of enterprise modeling languages have been elaborated world-wide. These developments have led to consider now that there are too many heterogeneous modeling languages available.

[12] present an important overview of methods, techniques, and tools used in Business Process Re-engineering (BPR). This work gives a list of some related business process modeling techniques and tools. it has been the starting point of the research presented by [1] , which gives a more thorough overview with detailed analysis of the mentioned techniques in [12] and others. [1] concludes that Business process modeling is a much-researched field but is neither well structured nor classified.

Main problems related to this situation are [6]:

- difficulties (impossibility in some cases) to transform one model designed using a language to a model expressed in another one;
- difficulties for an enterprise to use a software tool if it is based on languages which are different from the ones adopted by the enterprise;
- difficulties for analysts to know and understand all the languages on the market.

Thus, it is not easy to choose the more efficient language for a specific problem. In a recent work, [11] enumerate many tools and methods for the industrial systems analysis and modeling by showing that if each one provides solutions to several problems, none is sufficient to analyze and model complex systems. We find these same problems with the hospital systems. The traditional approaches seem often too abstract being used on the hospital field or only conceived for a specific problem [8] [14].

In his work, [7] gives some great characteristics of the hospital systems by reminding that the human element is very important and difficult to take into account, because of each patient uniqueness and on the sometimes unforeseeable patient pathways, but also because of medical staffs autonomy in their working methods.

[2] analyzes, through the study of representative models, the evolution of incorporating the concept of human resources and its links, in business models. We note that if some languages, such as MECI (Modélisation d'Entreprises pour la Conception Intégrée, 'Companies Modeling for Integrated Design') [16] consider the competencies concepts; but do not consider the complex combinations of human resources, such that it is present in the hospital system (combinations of more than two or three operators with complex rules of allocation in terms of preference, priority...).

Thus, in spite of the numerous languages and modeling tools available, we did not find any specification tools that allow us to take into account all the specificities of the studied systems (hospital systems) with the level of detail wished for simulation (microscopic level corresponding to the patient).

## 3. DEFINITION OF A METHODOLOGY AND AN ANALYSIS LANGUAGE: "LAES"

The choice of a method or a language depends on the modeling objectives but also on the constraints which can be posed in term of accessibility, ease of use, training times, intrinsic to each one of these methods. In our work, the main goal is to design and to provide hospital teams with decision-making tools making it possible to cover the various temporal horizons. We have two important constraints:

- a constraint related to the systemic complexity of the studied systems;
- a constraint with the diversity of the participants on the project which requires to find methods and tools easy to acquire, so that the users can constantly understand and possibly complete the formalized knowledge (for nursing staff and doctors) and to reflect these modifications on the developed tool (for computer scientists).

We use the ASDI methodology (Analysis, Specification, Design and Implementation) worked out and developed initially by Gourgand [9] and officially presented at Tours (France) and Reno (United States) [10]. Largely used, by the LIMOS, but also in other research teams (Nancy, La Rochelle…), it makes it possible to design a modeling methodology of a class of systems, the generic knowledge model of this class, and to carry out a software components library which is exploited to generate action models (data-processing programs) for a system of the class. This methodology was designed following many studies concerning the modeling and the evaluation of complex systems (industrial systems, transport systems…). We are particularly interested in two modeling approaches:

- an object-oriented approach for the systemic breaking down of the system in physical subsystem (entities), logical subsystem (flows) and decisional subsystem (management rules) and the taking into account of the human resources activities not directly linked to the patient pathway: We use UML (Unified Modeling Language) [15] which has the advantage of being a very succeeded formalization and easily comprehensible by "not initiated". We used the classes diagrams and the activity charts which are not developed in this paper;
- a process oriented approaches to model the patient pathways. We choose, initially, ARIS language (http://www.ids-scheer.com).

With the aim of formalizing the knowledge on a hospital structure and running, we carried out an information collection from the various departments. This collection enabled us to identify the significant number and the complexity of the patient pathways with the use of probabilities and/or management rules in these pathways and in the "elementary operations" (the more little component of the activity) which compose them: boolean expressions on the resources, priorities, preferences…. We did not succeed in modeling this complexity with ARIS language, and thus proposed the use of a new language. LAES is a Language of Analysis and Evaluation of Systems (LAES) suggested a few years ago by the LIMOS to model the complex systems based on a transaction approach.

We define the transaction approach as the description, in the formalism chosen, of the system operations while specifying, for each type of entities flow, the routing of these entities and their successive treatments: the system is described by the movement of the service requests in the system. This approach is used in the hospital systems since one is interested in the patient pathways, i.e. with the move of the customers through the system; it is also the most complex to model.

The objectives of LAES are [3] [4]:

- to allow the structure representation and the system running and design a knowledge model by using a transaction approach;
- to store the model obtained in the form of a file says "running file" by using a simple description language;
- to exploit the running file by a software which creates the action model and possibly do evaluation;
- to allow the use of evaluation tools (QNAP2, SIMAN…);
- to provide a specific evaluation tool for analytical resolution.

One of the advantages of LAES is its simplicity but also its flexibility which makes it possible to consider extensions to adapt it, as well as possible, to the system studied.

### 3.1 LAES

#### *3.1.1 Basic concepts of LAES: the stage and the path*

The knowledge model is built on two levels: the first level, known as overall, uses the stage and path concepts, and the second level, known as detailed, gives the contents of each stage. The studied systems are supposed to be open. For the class of the studied systems, several service requests are accepted jointly. A service request corresponds to a customer category asking for a set of elementary services. The advance of the customers is represented in a deterministic way. By a stretch of language, we confuse the category, the customer category and the customer.

In the request processing, various situations can occur: unsatisfied request, complete processing. By definition with each one of these cases correspond a path. A path is thus the route that a customer will take from its input to its output. A route is a sequence of stages, a stage being a set of elementary operations. A stage can be common to several paths. These concepts of stage and path appear natural for the studied systems. Indeed, two identical requests do not have the same probability of succeeding. On the other hand these concepts are not rigid since for the same system the selected level of modeling can be more or less fine.

The stage "output" corresponds, in general, to the realization of a condition (availability or not of the requested resource, external event, processing end …) and the stage "input" corresponds to the realization of an event (customer arrived, a passive resource allocation …). In the case of a passive resource request, two cases can occur: the resource is or is not available. There are thus an output of the request stage (i.e. end) and input in the allowance stage or in the stage corresponding to the lack of resource and in both cases, the continuance of the processing.

### *3.1.1.1 The overall graphic*

The overall graphic representation is a tree structure revealing the various paths which can take a customer and the stages constituting the paths. The root is input stage of the system. The sheets of the tree structure are obligatorily the output stages of the system. Another node than the root cannot be an input stage. A path is the sequence of stages which connect the input stage and an output stage. The Figure 1 represents three paths. OUT indicates outside (input or output of the system). The stages are numbered from top to bottom and from left to right from number 1. A customer will take path 1 or path 2 or path 3 (the "or" being exclusive).

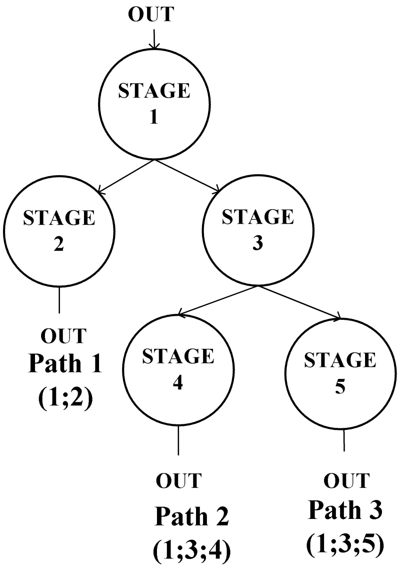


**Figure 1. Example of overall graphic representation**

### *3.1.1.2 The detailed graphic representation*

Each stage of the overall graphic representation is the subject of a detailed chart. This one gives the sequence of the elementary operations and the symbols of structuring: stage beginning and end; elementary operation; waiting delay; passive resource seizing and release; loops; parallel processing; stage call and sub stage call. For a complete description of LAES, see [4].

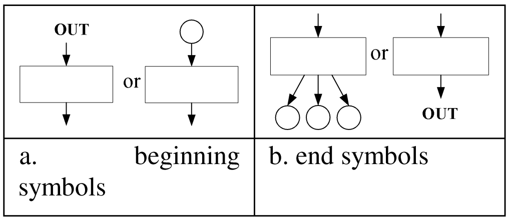
In LAES, we distinguish two types of resources:

- the active resources which carry out an elementary processing;
- the passive resources which are necessary to the active resources and which show by their occupation, the state of the request progress. They do not carry out elementary processing.

*Beginning and end of stage*

The beginning of a stage is indicated by the number of the preceding stage or the outside and the end of a stage is indicated by the number of the following stages or the outside (Figure 2).

The symbol which follows the beginning or which predates the end of a stage is an elementary operation symbol or a structuring symbol.



a. beginning symbols

b. end symbols

**Figure 2. Symbol of the beginning and the end of stage**

*Execution of an elementary service or operation*

The execution of an operation by an active resource is represented by (Figure 3), where:

- i is the number or the name of the active resource;
- t is the time of the elementary operation;
- c concerned customer class;
- n is the quantity of executions (optional if n =1);
- b is the bit of setting in waiting queue which is equal to 1 if it must be put in the waiting queue, 0 if not (optional if b =1).

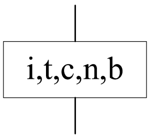
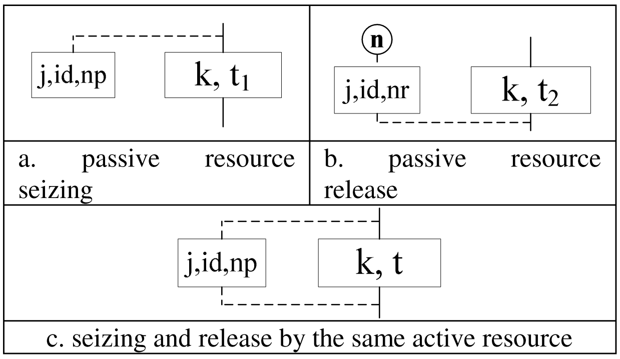


**Figure 3. Symbol of the elementary operation**

*Passive resource seizing and release*

The seizing and the release of passive resource are compulsory carried out by an active resource. Figure 4a represents the seizing of the passive resource *j* by the active resource *i*. The seizing starts with the execution of the elementary operation. The seizing of a passive resource in a stage implies the release of this same resource in all the paths containing the stage of the seizing. The Figure 4b represents the release of the passive resource *j* by the active resource *i*. The release follows the execution of the elementary operation. *n* indicates the number of the stages where the seizing was carried out. A passive resource can be seize and release by the same active resource (Figure 4c). Passive resources seizing and releases must be identified. One uses for that an identifier of seizing *id* which follows the number or the name of the passive resources *j*. If several passive resources seizing or several passive resources releases belonging to the same station are carried out simultaneously, the number of seizing *np* and the number of release *nr* are indicated after the seizing identifier.



a. passive resource seizing

b. passive resource release

c. seizing and release by the same active resource

**Figure 4. Symbols of the passive resources seizing and release**

*Loop, Stage and Sub stage call*

LAES use loops. Their structure rules are those applied into algorithmic languages. In the stage description we can find the whole or a part of other stage content. Rather than to reproduce these contents, we can use a stage or a sub stage call.

*3.1.2 Basic principles of LAES : LAES Codes*

Each element constituting the diagrams corresponds to a code of four letters and a list of attributes (Table 1).

**Table 1. LAES Codes**

| LAES Code | Definition | Sentences |
|---|---|---|
| **APPH** | Stage call | N, APPH, NP, < list >; |
| **APSP** | Sub stage call | N, APSP, NP, < list >; |
| **BLOC** | Block definition | N, BLOCK, NB, NAME, NC; |
| **CATE** | Definition of the numbers of category | N, CATE, NC, < list >; |
| **CHEM** | Path description | N, CHEM, NCH, NAME, NCA, < proportion >, NPH, < list >; |
| **CLAS** | Definition of the category classes | N, CLAS, NC, NCL, < list >; |
| **CLNT** | Customer description | N, CLNT, NCA, NAME, < proportion >, NCH, < list1 >, NR, < list2 >; |
| **DEBO** | Loop beginning | N, DEBO, < number >; |
| **DEFC** | Constant definition | N, DEFC, NAME, numerical value; |
| **DEFV** | Variable definition | N, DEFV, NAME, arithmetic expression; |
| **DEPA** | Parallel processing beginning | N, DEPA, ND, NT; |
| **DEPH** | Stage beginning | N, DEPH, NP, < number >, < list >, < ind >; |
| **DEPR** | Passive resource seizing beginning | N, DEPR, NR, < number >, ND; |
| **DESP** | Sub stage beginning | N, DESP, NP, < list >; |
| **EXEC** | Elementary service | N, EXEC, NR, < time >, NC, < number >, < bit >; |
| **FIBO** | End of loop | N, FIBO; |
| **FIBR** | End of branch | N, FIBR, NB, ND; |
| **FINF** | End of file | N, FINF; |
| **FIPA** | End of parallel processing | N, FIPA, ND; |
| **FIPH** | End of stage | N, FIPH, NP; |
| **FIPR** | End of seizing | N, FIPR, NR, < number >, NF; |
| **FISP** | End of sub stage | N, FISP, NS; |
| **PHAS** | Stage definition | N, PHAS, NP, NAME, NC, NB, < list1 >, < list2 >…; |
| **RESA** | Active resource definition | N, RESA, NR, NAME, NB, < time >; |
| **RESP** | Passive resource definition | N, RESP, NR, NAME, NB; |
| **TEMP** | Waiting time | N, TEMP, < time >, NC, < bit >; |
| **TEXT** | Comment | N, TEXT, < text >; |

A structure makes it possible to join together these codes and to compose a file named "LAES running file”. It is composed of several blocks, each one of these blocks has a specific function:

- block 1: resources, categories and classes declaration;
- block 2: declaration of the customers, the paths and the stages;
- block 3: declaration of the variables;
- block 4: details of the stages of category 1;
- N+3 block: details of the stages of category N;
- N+4 block: sub stages details.

Figure 5 gives a simple example.

```
10, BLOC, 1, STRUC;
20, RESA, 1, RES1, 1, 5.0;
30, RESA, 2, RES2, 1, 5.0;
40, RESP, 3, RES3, 2 ;
50, CATE, 1, 1;
10, BLOC, 2, CCP;
20, CLNT, 1, ARRIV, 1.0, 2, 1, 2;
30, CHEM, 1, CHEM, 0.5, 2, 1, 2;
40, CHEM, 2, CHEM2, 0.5, 2, 1, 3;
50, PHAS, 1, PA1, 1, 2, 2, 3;
60, PHAS, 2, PA2, 1, 1, 0;
70, PHAS, 3, PA3, 1, 1, 0;
10, BLOC, 4, ARRIV;
10, DEPH, 1;
[…]
```

**Figure 5. Example of LAES running file (extract)**

## 3.2 Hospital systems specificities: extensions of the language in “LAESH”

In this sub section, we focus on the specificities of the hospital systems that we have considered in order to design the knowledge model. These specificities include the quantity of necessary human resources for an elementary operation which is generally fixed in a manufacturing system and rarely exceeds one to two operators for the same operation, whereas hospital systems can have up to six operators for the same elementary operation, along with complex management rules related to these operators.

Thus, the principal extensions introduced to the language relate to the resources whose management rules are often complex. The specificity of our customers, the patients, justifies the management rules of human resources more complex: for example, it is hard to imagine to let a pregnant woman alone in the corridor of an obstetric unit because the resources which "should" be allocated for her are not immediately available.

We proposed an extension of LAES with the Language of Analysis and Evaluation of the Hospital Systems (LAESH) which enables to formalize the patient movements through the system taking into account the specificities of such systems.

*Multiplication of the paths with each XOR on an elementary operation mobilizing the same resources*

The patient pathway includes many stages where we can have the choice between 2 operations (XOR) mobilizing the same resources (R1 and a Room) and we can have several sequences of this type in the same pathway. With LAES, each XOR results in the creation of numerous paths (Figure 6).

We propose an extension which makes possible to reduce all these paths in a single elementary operation where $p_i$ is the probability that the elementary operation *i* lasts the time $t_i$ and (1-$p_i$) is the probability that the elementary operation *i* lasts the time $t'_i$ (Figure 7).

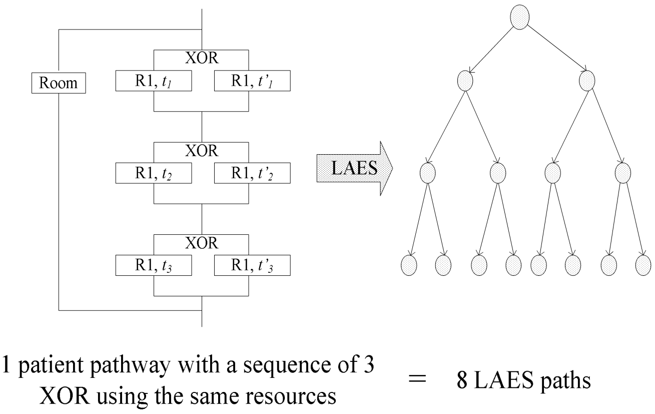


**Figure 6. Transition of a set of XOR conditions with identical resources in LAES**

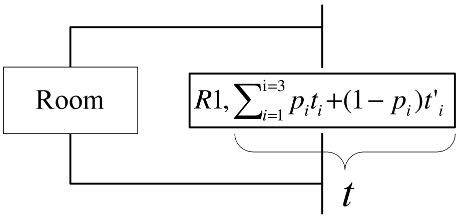


**Figure 7. Interpretation of the figure 6 with LAESH**

*Specific and cyclic mobilization of the active resources by a long elementary operation with a variable duration*

Some operations mobilize the passive resources during all the time *t'*, but mobilize the active resources only punctually. The active resources must thus be able to be released to carry out other operations simultaneously. In LAESH, we place a waiting delay of time *t* after the operation execution followed by a loop which makes it possible to start again the operation and waiting delay *NB* times (Figure 8). *t* can be a random variable.

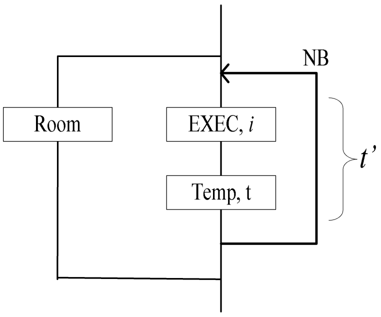


**Figure 8. Example of a waiting delay and a loop for a specific and cyclic operation**

*Use of several active resources for an operation with conditions "AND" (∧), "OR" (∨) and "XOR" (⊕)*

We consider that human resources (operators) are active resources in LAESH. Table 2 gives the representation in LAESH for various operator combinations where *k* is the number of active resources of each type.

**Table 2. LAESH representation of the operator combinations**

| Elementary operation provided by | Representation in LAESH |
|---|---|
| k resources of the type i | EXEC, *i*, t, c, n, b, k |
| $k_i$ resources of the type i **AND** $k_j$ resources of the type j | EXEC *i* **AND** *j*, t, c, n, b, $k_i$, $k_j$ |
| $k_i$ resources of the type i **XOR** $k_j$ resources of the type j | EXEC *i* **XOR** *j*, t, c, n, b, $k_i$, $k_j$ |
| $k_i$ resources of the type i **OR** $k_j$ resources of the type j | EXEC *i* **OR** *j*, t, c, n, b, $k_i$, $k_j$ |

*Simultaneous use of a resource by several customers*

Some operations mobilize an active resource only punctually because this one is shared by several simultaneous operations (ex: anesthetist doctor shared between two operating rooms).

LAESH extension: when an active resource can treat several customers of the same or different categories, simultaneously (the active resource is shared between several customers):

**EXEC, i, t, c, n, b, k, P(x)**

*x* is the maximum number of customers concerned with the sharing. This extension also was added to the passive resources (ex: to limit place in waiting room).

*Same active resources for the sequence of several operations: personalized resources*

Some operations mobilize the same type of resource and it is necessary, by defect and unless otherwise specified or pre-emption, that the active resource must be the same for the operations set. In LAESH we define the basic management rule according to which the resource of the type *i* or the resources combination used for a sequence of elementary operations remains the same one, unless otherwise specified or pre-emption.

*Pre-emption of a resource for the execution of several simultaneous elementary operations into: operation priority*

Some operations have priority compared to others for obtaining the resources; it must be able to preempt a resource already occupied on the execution of a less priority operation (Figure 9).

The priority task must be able to find, by defect, the resource which began the execution before the pre-emption (personalized resource). If this one is not available any more, it must be able to be completed with another resource of the same type.

LAESH extension: when it is necessary, we introduce a priority index into the elementary operation. An elementary operation (EXEC) has, by defect, the smallest priority (zero):

$$\mathbf{EXEC_{y}, i, t, c, n, b, k,}$$

where $y$ is priority index of the elementary operation.

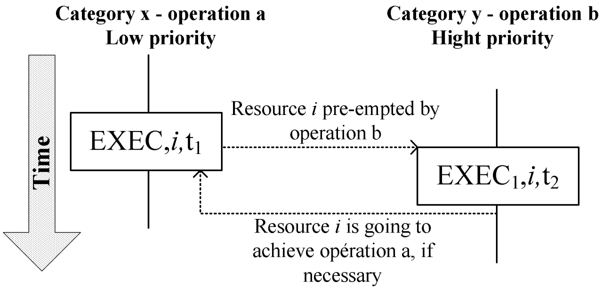


**Figure 9. Graphic interpretation of the operations priority on 2 categories**

### 3.3 Mapping from LAESH to the simulation tools

LAESH has a fundamental advantage for our project: is a good communication tool with the hospital teams (overall graphic representation) and with the software developers.

Table 3 gives the mapping rules from the knowledge model to the action model realized with the simulation software Witness (http://www.lanner.com/). We created a software components library to answer hospital systems specificities.

**Table 3. Transition rules from the knowledge model to the Witness action model**

| **Knowledge Model** | | **Action Model** |
|---|---|---|
| **UMLObject** | **LAESH** | **WITNESS** |
| Patient | Category | Type of article |
| Area | EXEC attribute | Module |
| Rooms | RESP, DEPR, FIPR | Machine |
| Human resources | RESA | Operator |
| ..... | EXEC, DEBO, FIBO | Cycle, Stock, Machine |
| | DEPA, FIPA | Attached resource |

## 4. APPLICATION TO AN OBSTETRIC UNIT MODELING

### 4.1 The knowledge model of the obstetric theatre with LAESH

We used LAESH to model the whole of the patient pathways of a future obstetric unit gathering two units now distinct. We give here an outline of the LAESH knowledge model. Table 4 gives the principal elements of the patient pathways.

Eight patient's categories were identified for the NHE obstetric unit. We distinguish the "fathers" processes which correspond to the main customers (patients) and the "sons" processes derived from the "fathers" processes (newborns).

- «Fathers» processes: emergency consultations with pregnant women; emergency consultations without pregnancy; delivery not included programmed caesarean; programmed caesarean; versions by external maneuvers; medical termination of pregnancy.
- «Sons» processes: Baby by single delivery; Baby by caesarean.

**Table 4. LAESH elements of the knowledge model**

| | | | |
|---|---|---|---|
| Patients categories | 8 | Type of Active Resources (staff) | 9 |
| Stages | 61 | Type of Passive Resources (rooms) | 31 |
| Patient pathways | 38 | Areas | 3 |
| Elementary operations | 153 | | |

Figure 10 gives an example of overall graphic representation which comprises 15 stages and 9 paths.

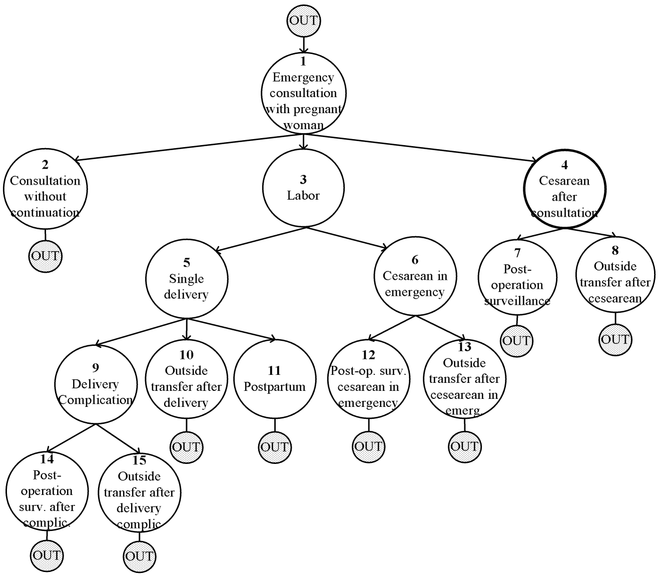

OUT
1 Emergency consultation with pregnant woman
2 Consultation without continuation
OUT
3 Labor
4 Cesarean after consultation
5 Single delivery
6 Cesarean in emergency
7 Post-operation surveillance
8 Outside transfer after cesearean
OUT
OUT
9 Delivery Complication
10 Outside transfer after delivery
11 Postpartum
12 Post-op. surv. cesarean in emergency
13 Outside transfer after cesearean in emerg
OUT
OUT
OUT
OUT
14 Post-operation surv. after complic
15 Outside transfer after delivery complic
OUT
OUT

**Figure 10. Category 1 - Emergency consultations with pregnant women**

Figure 11 gives the detailed graphic representation of stage 4. Figure 12 gives the operations sequence of the sub stage 3 called by stage 4. To answer hospital systems specificities, such as the «obligatory» execution of some operation, whatever the available resources (emergency, complication, etc.) the LAESH knowledge model can take into account for each elementary operation, a boolean expression on the types of resources and allows the concept of preference. Figure 13 represents an elementary operation of sub stage 3. This operation takes place in the operating area (OA). Based on the priority rules, it will call primarily the operating area human resources or the multi areas resources before calling, if necessary, the next area resources: childbirth area (CA). In the following example we see that are initially called operating area midwife (OAMW) or multi-midwife (MW), before calling, as a last resort, childbirth area midwife (CAMW).

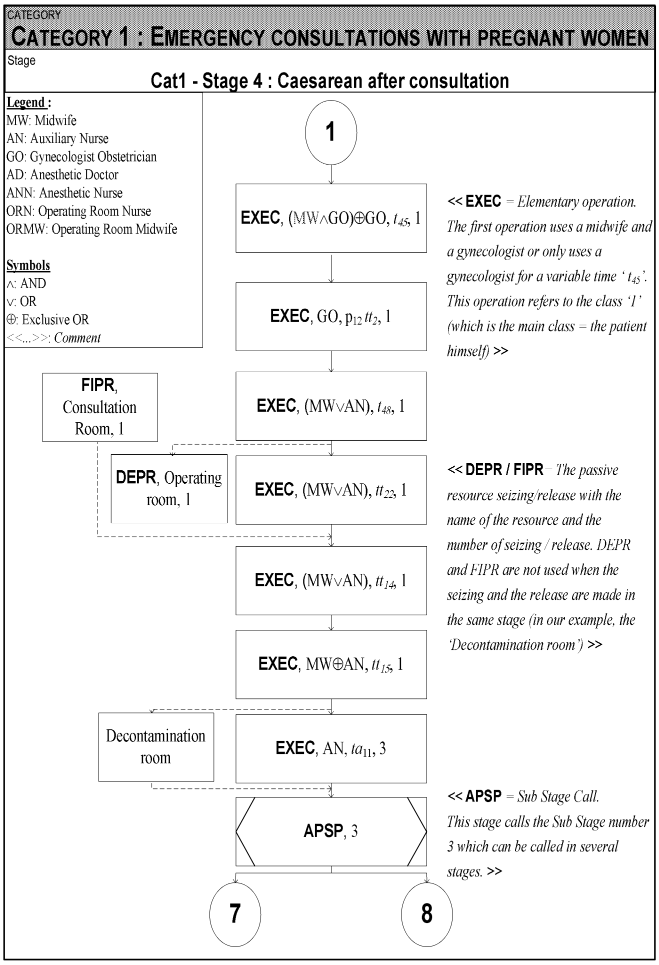

CATEGORY
CATEGORY 1 : EMERGENCY CONSULTATIONS WITH PREGNANT WOMEN
Stage
Cat1 - Stage 4 : Caesarean after consultation
Legend :
MW: Midwife
AN: Auxiliary Nurse
GO: Gynecologist Obstetrician
AD: Anesthetic Doctor
ANN: Anesthetic Nurse
ORN: Operating Room Nurse
ORMW: Operating Room Midwife
Symbols
∧: AND
∨: OR
⊕: Exclusive OR
<<...>>: Comment
1
EXEC, (MW∧GO)⊕GO, t45, 1
<< EXEC = Elementary operation. The first operation uses a midwife and a gynecologist or only uses a gynecologist for a variable time 't45'. This operation refers to the class '1' (which is the main class = the patient himself) >>
EXEC, GO, p12 tt2, 1
FIPR, Consultation Room, 1
EXEC, (MW∨AN), t48, 1
DEPR, Operating room, 1
EXEC, (MW∨AN), tt22, 1
<< DEPR / FIPR = The passive resource seizing/release with the name of the resource and the number of seizing / release. DEPR and FIPR are not used when the seizing and the release are made in the same stage (in our example, the 'Decontamination room') >>
EXEC, (MW∨AN), tt14, 1
EXEC, MW⊕AN, tt15, 1
Decontamination room
EXEC, AN, ta11, 3
<< APSP = Sub Stage Call. This stage calls the Sub Stage number 3 which can be called in several stages. >>
APSP, 3
7
8

**Figure 11. Category 1- Phase 4: Detailed graphic representation**

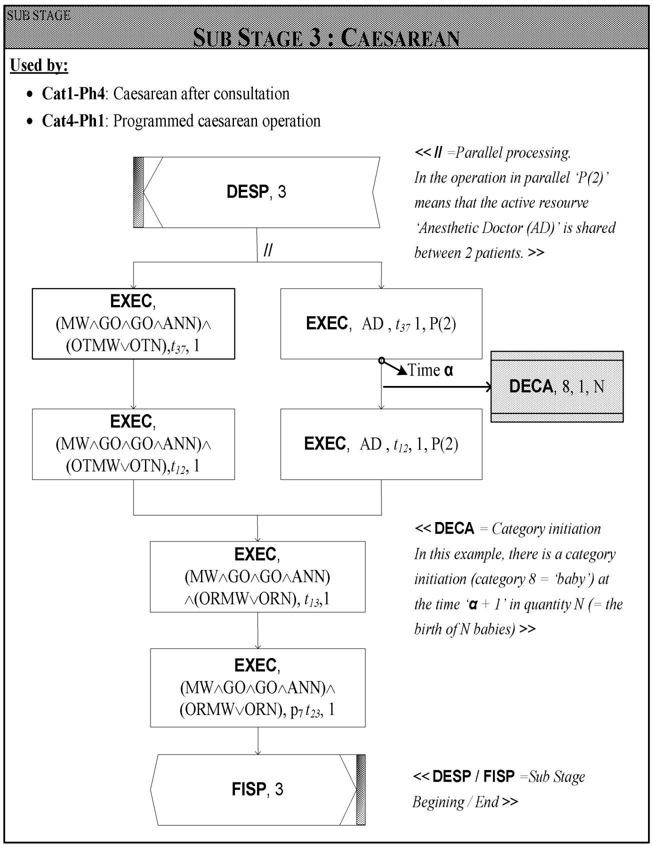

SUB STAGE
SUB STAGE 3 : CAESAREAN
Used by:
Cat1-Ph4: Caesarean after consultation
Cat4-Ph1: Programmed caesarean operation
<< // =Parallel processing. In the operation in parallel 'P(2)' means that the active resourve 'Anesthetic Doctor (AD)' is shared between 2 patients. >>
DESP, 3
//
EXEC, (MW∧GO∧GO∧ANN)∧ (OTMW∨OTN),t37, 1
EXEC, AD , t37 1, P(2)
Time α
DECA, 8, 1, N
EXEC, (MW∧GO∧GO∧ANN)∧ (OTMW∨OTN),t12, 1
EXEC, AD , t12, 1, P(2)
<< DECA = Category initiation In this example, there is a category initiation (category 8 = 'baby') at the time 'α + 1' in quantity N (= the birth of N babies) >>
EXEC, (MW∧GO∧GO∧ANN) ∧(ORMW∨ORN), t13,1
EXEC, (MW∧GO∧GO∧ANN)∧ (ORMW∨ORN), p7 t23, 1
FISP, 3
<< DESP / FISP =Sub Stage Begining / End >>

**Figure 12. Sub stage 3 – Caesarean**

**EXEC**, (MW AND GO AND GO AND AN) AND (OTMW OR OTN),T$_{37}$, 1

Legend :
MW : Midwife
GO : Gynaecologist Obstetrician
AN : Anaesthetic Nurse
OTN : Operating Theatre Nurse
OTMW : Operating Theatre Midwife

**Figure 13. Elementary operation**

Active resources boolean combining of elementary operation described with the Figure 13 is reflected in the following way in a simulation model developed with Witness 2006 (which does not handle the OR operator) :

**(OAMW AND GO#2 AND AN) AND (OTMW AND OTN)**
**XOR**
**(MW AND GO#2 AND AN) AND (OTMW AND OTN)**
**XOR**
**(OAMW AND GO#2 AND AN) AND OTMW**
**XOR**
**(MW AND GO#2 AND AN) AND OTMW**
**XOR**
**(OAMW AND GO#2 AND AN) AND OTN**
**XOR**
**(MW AND GO#2 AND AN) AND OTN**
**XOR**
**(MWCA AND GO#2 AND AN) AND (OTMW AND OTN)**
**XOR**
**(MWCA AND GO#2 AND AN) AND OTMW**
**XOR**
**(MWCA AND GO#2 AND AN) AND OTN**

To complete these patients pathways and identify the various tasks assigned to the whole of person intervening in the system, we have for each staff category (midwife, helps healthcare, etc.) detailed planning with all tasks done on the day, and which are not totally dependent of the patient (rooms preparation, training, administrative tasks, etc.). The knowledge model is generic with many elements user-definable: resources type and number; classes, pathways and inter-phases probabilities; time and priorities of elementary operations; arrival laws of flow entities (patients).

## 5. CONCLUSION AND PROSPECTS

The obstetric theatre simulation model, designed starting from Witness, is operational today. It was presented at hospital and was validated. This tool is implemented in the obstetric units. The doctors and nurse staff can test different scenarii from organization and note their incidences on the future system. This decision-making tool was presented to the 33rd International Conference on Operational Research Applied to Health Services (ORAHS) and in several medical conferences. LAESH made it possible to have a common language between hospital teams and software developers.

Further work concerns mainly the use of LAESH in other healthcare departments (pediatric emergency…) and the comparison of the results obtained with different simulation software.

## 6. REFERENCES

[1] Aguilar-Savén, R.S. (2004) 'Business process modelling: Review and framework', *International Journal of Production Economics*, Vol 90 (2), 129-149.

[2] Bennour, M. (2004) '*Contribution à la Modélisation et à l'Affectation des Ressources Humaines dans les Processus'*, PhD thesis, University of Montpellier II.

[3] Chabrol, M. (1986), '*Développement et utilisation de QNAP2 pour l'évaluation des performances par modèles analytiques'*, PhD thesis, University of Blaise Pascal, Clermont-Ferrand.

[4] Chabrol, M. Gourgand, M. (1991), 'Software environment for queueing network modeling', *In: Proceedings of 2nd International Conference RRES 91*, Milan, Italy.

[5] Chabrol, M. Gourgand, M. Rodier, S. (2008) 'A modeling methodology and its application to the design of decision-making aid tools for the hospital systems', *In: Proceedings of IEEE International Conference on Research Challenges in Information Science (RCIS),* Marrakech, Morocco, 161-172.

[6] Chen, D. Vallespir, B. Doumeingts, G. (2002) 'Developing an unified enterprise modelling language (UEML)–Roadmap and requirements', Collaborative Business Ecosystems and Virtual Enterprises, I*n: Proceedings of third IFIP Working conference on infrastructures for virtual enterprise, PROVE*, Sesimbra, Portugal.

[7] Combes, C. (1994) *'Un environnement de modélisation pour les systèmes hospitaliers'*, PhD thesis , University of Blaise Pascal, Clermont-Ferrand.

[8] Galland, S. Grimaud, F. Beaune., P Campagne, J.P. (2003) 'MAMA-S: An introduction to a methodological approach for the simulation of distributed industrial systems', *Int. J. Production Economics*, vol. 85, 11-31.

[9] Gourgand, M. (1984) *'Outils logiciels pour l'évaluation des performances des systèmes informatiques'*, State thesis, University of Blaise Pascal, Clermont-Ferrand.

[10] Gourgand, M. Kellert, P. (1992) 'An object-oriented methodology for manufacturing system modeling', *Summer Computer Simulation Conference*, 1123-1128, Reno (US), 27-30 Juillet 1992.

[11] Hernandez-Matias, J.C. Vizan, A. Perez-Garcia, J. Rios, J. (2008) 'An integrated modelling framework to support manufacturing system diagnosis for continuous improvement', *Robotics and Computer-Integrated Manufacturing*, vol. 24, 187–199.

[12] Kettinger, W.J., Teng, J., Guha, S., (1997). Appendices for Business process change: A study of methodologies, techniques and tools. *Management Information Systems Quartely Archivist*, vol. 14 (1), Appendices 1–8.

[13] Mayer, R.J. Menzel, C.P. Painter, M.K. Dewitte, P.S. Blinn, T. and Perakath, B. (1995) 'Information integration for concurrent engineering (IICE)', *IDEF3 Process description capture method report*.

[14] Moreno, L. Aguilar, R.M. Pineiro, J.D. Estevez, J.F. Sigut, J.F., Gonzales, C. (2001) 'Using KADS methodology in a simulation assisted knowledge based system : application to hospital management', *Expert system with application* , vol.20, 235-249.

[15] Object Management Group (OMG). (2007).*'Unified Modeling Language, specifications'* v2.1.1.Needham, MA, U.S.A.: OMG.

[16] Pourcel, C. et Gourc, D. (2002) 'Modélisation d'entreprise : la méthode MECI', *Ecole de printemps*, Modélisation d'entreprise d'Albi-Carmaux.

[17] Roque, M. Vallespir, B. Doumeingts, G. (2008), 'Interoperability in enterprise modelling: Translation, elementary constructs, meta-modelling and UEML development', *Computers in Industry*, vol. 59, 672–681.

## BIOGRAPHY

**MICHELLE CHABROL** is Assistant Professor of Computer Science in the University of Blaise Pascal (Clermont-Ferrand, France). Her research and teaching interests include software engineering, system modeling and simulation.
chabrol@isima.fr

**MICHEL GOURGAND** is Professor of Computer Science in the University of Blaise Pascal (Clermont-Ferrand, France) and manages the research team Modelling and Decision Aid of the LIMOS (Laboratoire d'Informatique, de Modélisation et d'Optimisation de Systèmes). His research and teaching interests include manufacturing system modelling, scheduling problems, supply chain management and metaheuristics.
gourgand@isima.fr.

**SOPHIE RODIER** is PhD student in Computer Science in LIMOS at the University of Blaise Pascal in France. Her research interests include modeling, simulation and optimization methods (heuristics, …) for health-care systems.
rodier@isima.fr.

# Methodological approach and decision-making aid tool for the hospital systems: Application to an emergency department

Julie Chauvet
LIMOS CNRS UMR 6158
Blaise Pascal University
63173 Aubière, France
(+33)4 73 40 04 37
chauvet@isima.fr

Michel Gourgand
LIMOS CNRS UMR 6158
Blaise Pascal University
63173 Aubière, France
(+33)4 73 40 75 14
gourgand@isima.fr

Sophie Rodier
LIMOS CNRS UMR 6158
& University Hospital of Clermont-Fd
63058 Clermont-Fd Cedex 1, France
(+33)4 73 75 04 37
rodier@isima.fr

**KEYWORDS**
Modeling methodology, decision-making aid tool, simulation, hospital systems, emergency department.

## ABSTRACT

The focus of this article is to present a methodological approach and its application for the design of decision-making aid tools dedicated to the hospital systems. This methodology has been developed to design decision making aid tools based on various resolution methods (mathematical formalization, simulation, etc…), which all begin by the formalization of a knowledge model (knowledge formalization) related to the studied system. We present the different steps to follow from the knowledge model to the design of a decision making aid tools based on a simulation model. A tool for a pediatric emergency department is presented and we give the results of the different tested scenarios.

## 1. INTRODUCTION

The main issues that impact hospital systems are similar to those presented by manufacturing systems and are primarily related to their sizing, to the understanding of their mechanisms, to the improvement of their productivity and to their performances evaluation. The hospital systems must have business logic and develop new management methods. Hospitals have to consider a more rigorous control of their processes to limit their expenditure and to optimize their organization. This change is particularly noticeable in the European continent where many countries are very attached to the public health and to the public utility. With an increased mobility and the border opening, the public hospital must be competitive as the private one. In the same time the population is more and more exacting in terms of care quality. To be helped in this step of performance evaluation and optimization, the health institution need decision-making aid adapted to their specificities. To design such tools, it is necessary to think about modeling approaches and methods. It is an essential step for the knowledge formalization of any system. The methods, tools and levels of detail used for modeling depend primarily on the studied system and the appointed objectives.

In this paper, we propose a methodology in order to design decision-making tools for an existing system or a system to come (for example an hospital under construction) and which cover all the temporal horizons: strategic, tactical and operational. The main aim of these tools is to be able to act (i) upstream, on the dimensioning of the structure and the staffing requirements, on the choice of management rules in terms of planning and resource allocation, of flows managements (human, material, information); (ii) during the system working to change and adapt the management rules and the resource allocation; (iii) downstream, to evaluate the system performances. Formalizing the knowledge and designing adapted tools, we chose to follow ASDI methodology (Analysis, Specification, Design, Implementation) [11]. We have applied our methodology on several systems: obstetrical unit, care unit, hospital porters department. In this paper, we have chosen to present its application to a pediatric emergency department in order to design a decision-making tool for the hospital managers.

Difficulties in the emergency department are mainly patient flows subject to many risks, low anticipation of the process of patient care, lack of control of the upstream and downstream of the patient pathway, the multiplicity stakeholders. The main objective of an emergency department is therefore to ensure rapid and qualitative care for the patients while planning the resources of the hospital system.

After a brief state of the art on the studies related to the emergency departments and on the methods and tools used, we propose a methodology for the hospital systems modeling and present the decision-making aid tool for the modeling of the emergency department of a French hospital.

## 2. STATE OF THE ART

In the literature, the main studied problems of the emergency department concerns modeling, simulation and optimization of the patient flow. The objectives may be to reduce the transit time of patients, to design and implement a tool for sizing and managing systems. Given the multiplicity of issues related to the optimization of emergency services, we present a grid (Table 1). It classifies each method resolution (modeling, simulation, mathematical approaches) and gives the various issues resolved.

Later in the paper, we take different sections of the reading grid that we explain the following data collection and modeling of the emergency department, action and results models. This presentation was done following the ASDI methodology (Analysis, Specification, Design, and Implementation) [11] which is presented in the next section. This approach is based on the design of a successive knowledge model and an action model through bricks software designed by a process of collecting management knowledge.

## 2.1 Data collection and modeling of the emergency department

### *2.1.1 Data collection from emergency department*

Collecting data necessary for various studies, several methods are used as observation ([18] [25][17][22]), group work or interviews ([33] [28] [26] [18]), derived from historical data ([29] [16] [18] [35]). In [28] and [18], information is collected through exchanges among medical and paramedical staff (for example, department heads, doctors, executives of health, auxiliary nurse…) of administrative and technical staff (example: stretcher). In the same way in [35], the data are collected through interviews with officials from the Emergency Department and observations of events on the field. The first method of gathering information, through an exchange with several people, is based on data recorded over a period of less than one year concerning the quantity of doctors, pharmacists, nurse, administrative staff and their schedules. The second method is to select and collect events unfolding in the emergency department over a period of 24 hours on different days. The observation periods may vary for example between several days [35] and several weeks [25]. [17] do not specify the duration of their comments, but they detail the subject of their comments. They study the patient pathway, the activities of human resources, and the occupation of physical resources. This analysis enables them to understand the reality of how the system works and to show its complexity. Other authors ([30][16]) propose to generate data using various tools such as Excel VBA.

After collecting information on the system, digital data are stored using tools such as Excel or in databases such as Access [25].

**Table 1. Main topics of study related to the simulation and the optimization on the emergency department**

| Problems | | | Authors |
|---|---|---|---|
| The modeling, the simulation and the optimization of the patient flows | Modeling | Patients pathway modeling with Petri nets | [13][14][24] |
| | Simulation | Elimination of the bottlenecks | [8][22] |
| | | Optimization of the patient flow | [17] |
| | | Determination of the quantity of beds, resources and patients | [34][21][28][30][2] |
| | | Determination the staffing requirements | [3] |
| | | Reduction of the waiting time for the patients | [27][20][26][16][18][15][9] |
| | | Decision-making aid in regard to the behavior of the system | [19][32][23][29][12][25] |
| | Mathematics problems | Determination of the optimum resource planning | [31][35][3] |

### *2.1.2 Modeling of the emergency department*

In order to understand an existing system or to come, it is important to model the system. This model may be: simple ([18][15]) or complex ([26]).

[14] have chosen to develop their model from the colored Petri nets. The interest of these Petri nets, interpreted and P-time resident is their readability and their aspects "communicating" for the members of a project. They also appear to be a tool for simulation processes that model. [24] propose to develop a model by providing some initial changes to the MOP so as to include specific service studied. The main contribution of this work the initial model is the introduction and definition of colored marks and data to feed stochastic simulation.

In [35] the emergency department is described with a simple flow model containing the main care stages of a patient (i.e. from its entry into service until his exit through the yard), the choice of the room (emergency room or bandage room), the diagnosis, the tests and the observations. These models can be compared with DFD (Diagram Flow Data) SSADM (Structured System Analysis and Design Method). Each phase is described separately the model by comments and figures as the quantity of patients per hour. ([2][12][25][17]) also represent the emergency department with a flow model, but they do not choose the same level of analysis.

Other authors ([28]) propose to describe the emergency department with the help of process diagrams representing the patient pathway. These diagrams are alike the diagrams activities UML2 ([1]). For each category of patients, a diagram of the process was designed. It contains the duration of each component of the process and the frequency of each connection between the different elements.

According to the approach chosen by ([21][9][22]), a conceptual model must be done, it should not be just a map of process or a flow diagram. The models seem from the formalism of Adonis [10]. For example, a map of the process must contain detailed descriptions and management rules associated with objects map process. It helps to understand the system before further progress in the project and therefore control the feasibility of the project. [16] also uses a map process with Microsoft Visio to analyze the different flows through the emergency department. It also determines the management rules, including the constraints related to resources. He questioned the level of detail that defines modeling by comparing both the time of the project at its value. He suggested to model the flow at a high level as a first step and then add details if necessary. [34] describe the process flow at macroscopic level. They state that this level of detail is appropriate for the design stage.

## 2.2 Action and results models of the emergency department

After presenting the data collection and different models, we compared the patterns of action and results of the emergency department of the grid (Table 1) following Table 2. We chose an article for each issue.

This choice was established following the issue of the article should correspond to all the benchmarks to evaluate all the items. The benchmarks used are as follows: Objectives of the study; Types of models; Tools simulation / resolution; Input data / output data; Conclusion / Remarks.

The study of the literature shows the dedicated problems, the lack of genericity of models and methodology and the level of accuracy sometimes little high for these models. A complete state of the art can be found in [7].

We propose a methodology and a single tool based on the discrete event simulation which allows giving some results for the different problems studied by the authors quoted in the table 1 in the steps of modeling and simulation.

This methodology is applied in the context of our study on the NHE (New Hospital d'Estaing, opening: march 2010) of Clermont-Ferrand (France). The pediatric emergency department of NHE will host children under better conditions than those proposed by the pediatric emergency department of the “Hôtel Dieu”, the current hospital. The activities of medical and paramedical staffs related to treatment of patients will not change but the organization of these activities will be evolved. Changes related to the physical structure must be taken into account as the quantity of care rooms. Among these changes, the pediatric emergency department of the NHE may be equipped with an additional unit: the HUSD (Hospital Unity of Short Duration), and a new post: the orientation nurse. This nurse will be in charge of the coordination of the patients flow. The objective of this function is to reduce waiting times, improve patient care as soon as he arrives in the emergency department, to define priorities (such vital emergency) and alternatives care. It has an impact on the transit time of the patients with a global vision of the situation.

**Table 2. Comparison of simulation models and mathematical (extract)**

| **Authors** | [31] | [3] | [30] |
|---|---|---|---|
| **Objectives of the study** | Development of an agenda for the planning of emergency doctor combining multiple constraints for a planning horizon of one month | Determination of the staffing requirements (simulation model)<br>Determination of the optimum resource planning (mathematical model) | Determination of the quantity of emergency rooms, pediatricians, intern, surgeons, specific rooms, stretcher in order to reduce the waiting time for patients |
| **Types of models** | Mathematic model | Simulation model /Mathematic model | Simulation model |
| **Tools simulation / resolution** | Cplex solver | Arena / Cplex solver | Arena |
| **Input data / Output data** | Quantity of resources<br>Hierarchical level<br>Quantity of days | Quantity of resources (simulation model)<br>Quantity of nurses by post (mathematical model) | Input data: quantity of patients<br>Output: time spent waiting in the emergency department, the time spent to get a available room |
| **Conclusion / Remarks** | Specific problem of planning | Specific problem of planning | Specific problem of design of a physical structure |

## 3. OUR METHODOLOGICAL APPROACH AND THE TOOLS USED

### 3.1 The ASDI Methodology [11]

First, the ASDI methodology recommends the construction of a generic knowledge model of the system class. The knowledge model is a formalization of the structure and operations of the system through a natural or graphic language. This generic model must be able to be instanced for any system of the class. So, we can design the knowledge model of our system from the generic knowledge model of the system class. This first work is the result of the Analysis and the Specification. This generic model must also allow the design of software components which will be used during the design stage of the action models development.

In a second step, the action model design is a translation of the knowledge model in a mathematical formalism (for example an analytical method which would exploit a queuing network analysis, or a data-processing model) or in a programming language (for example a discrete event simulation model).

Some user interfaces can be added. This model has to be computer readable and to provide some performances criteria about the system, whose analysis can provide enough information to proceed to a real action on the system, and thus on the knowledge model (Figure 1). This work is the result of the Design and the Implementation.

Initially used to design a modeling environment for the manufacturing systems, ASDI was taken again and adapted for other systems such as urban traffic systems with ASDI-mi, which introduces multiple and incremental modeling concepts [4]. In 2006, [5] take an interest in the coupling of financial flows with physical flows and propose a generic conceptual approach for the modeling of Supply Chains by giving its implementation for the Hospital Supply Chain systems (ASDI-sch).

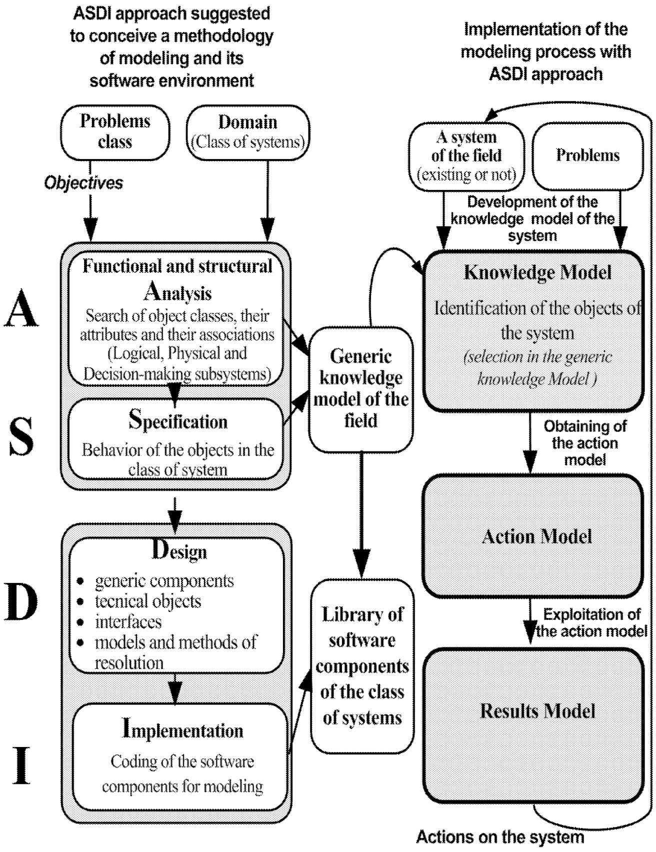

ASDI approach suggested to conceive a methodology of modeling and its software environment
Implementation of the modeling process with ASDI approach
Problems class
Domain (Class of systems)
Objectives
A system of the field (existing or not)
Problems
Development of the knowledge model of the system
A
S
D
I
Functional and structural Analysis
Search of object classes, their attributes and their associations (Logical, Physical and Decision-making subsystems)
Specification
Behavior of the objects in the class of system
Generic knowledge model of the field
Knowledge Model
Identification of the objects of the system
(selection in the generic knowledge Model)
Obtaining of the action model
Action Model
Design
generic components
tecnical objects
interfaces
models and methods of resolution
Implementation
Coding of the software components for modeling
Library of software components of the class of systems
Exploitation of the action model
Results Model
Actions on the system

**Figure 1. ASDI Methodology**

### 3.2 The tools used

One of the advantages of the ASDI methodology is the possibility of using different tools for each step. In this paper, we present a decision making aid tool for a pediatric emergency department. This tool is based on a discrete event simulation model. The Figure 2 shows the languages, the tools and the computer software which were used for each step of the ASDI Methodology in order to design a decision making aid tool for the emergency department of the NHE.

In order to provide an appropriate answer to the systemic complexity of the hospital systems, we used various modeling tools based on two complementary approaches:

- An object-oriented approach for the systemic breakdown of the system in physical, logical and decision-making subsystems, and for the consideration of the human resources activities which are not directly linked to the patient care. We used the Unified Modeling Language (UML) to formalize this knowledge, using different class diagrams for the systemic breakdown, and activity charts for the actors activities.
- A process oriented approach in order to model the patient pathways. We used the Language of Analysis and Evaluation of the Hospital Systems (LAESH) which enables to formalize the patient movements through the system. An LAESH representation models the patient pathway in two different levels: the global level uses the concepts of category, stage and path, and the second level, more detailed, describes precisely the contents of each stage. This language and its application are presented in previous works [6].

| Analysis | Specification | Design | Implementation |
|---|---|---|---|
| Systemic Breakdown, Human resources activities, Patient pathway | | Simulation Model, User interfaces | |
| UML / LAESH | | Witness / Excel | |

**Figure 2. Used tools and languages for the decision making aid tool design**

Several action models can be deduced starting from the same knowledge model depending on the formalism that has been chosen to build the action model and/or on the degree of accuracy established by an action model: this plurality of action models leads to a wide range of decision-making aid tools which are usually complementary. The choice of a particular type for the action model will depend on the tool objectives, along with the simplifying rules concerning the structure, the operating part of the system itself and the description of the system load. It also depends of course on the wished results. It is thereby extremely important to delimit the results that have to be obtained (and which will be presented in the results model). So as to fulfill user expectations we proposed an action model based on a discrete event simulation model. We used Witness software for the simulation model and Excel (at the request of users) for the user interfaces.

## 4. THE KNOWLEDGE MODEL OF THE PEDIATRIC EMERGENCY DEPARTMENT

The knowledge model is composed of three parts:

- The systemic break down which comprise all the entities of the system.
- The patient pathways which specify all the possible paths for the patient according to its pathology and gravity.
- The management rules which complete the previous parts.

### 4.1 Systemic breakdown

To obtain knowledge models of complex systems, ASDI recommends a systemic breakdown of the studied system in three communicating subsystems: (i) the Physical Subsystem (PSS) defines the physical entities set , their geographical distribution and the links with each other; (ii) the Logical Subsystem (LSS) represents the flows of entities which have to be handled by the system, along with the set of operations concerning these flows, and the nomenclatures which refer to this set; (iii) the Decision-making Subsystem (DSS) contains the working rules of the system. The Figure 3 represents the links between the three sub systems. The PSS of the emergency department is composed by nine classes and twenty two subclasses (break down); the LSS of the emergency department is composed by seven classes and two subclasses and the DSS is composed by five classes.

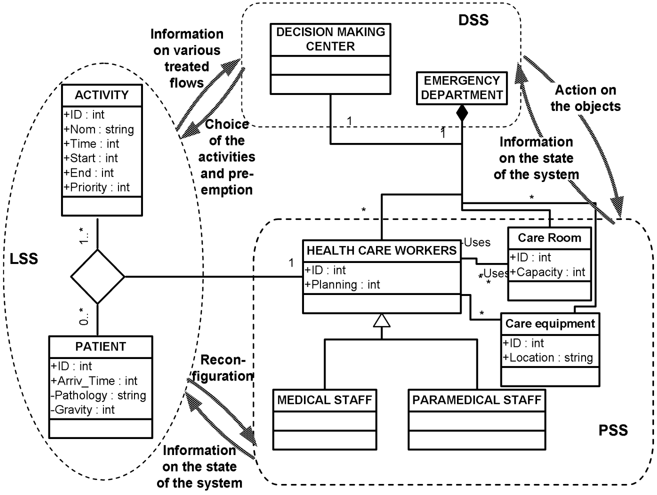


**Figure 3. The links between the three subsystems**

### 4.2 Patient pathways

At the NHE, the different examination rooms of the emergency department will be dedicated to some pathologies in order to make easier the staff work:

The pathology “suture”: two care rooms are dedicated to the injuries that are exposed due to broken skin (open wounds).

The pathology “fracture”: one care room is dedicated to the fracture. Once the fracture has been diagnosed, the initial treatment for most limb fractures is a splint. Padded pieces of plaster or fiberglass are placed over the injured limb and wrapped with gauze and an elastic wrap to immobilize the break.

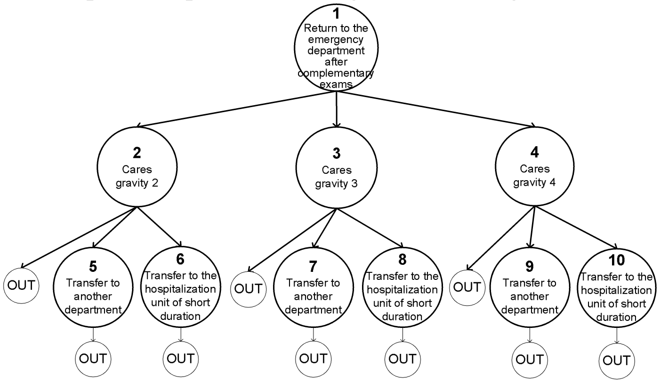
The pathology “psychological problem”: one care room is dedicated to the disturbed patients.

The pathology “vital emergency”: one care room is dedicated to the vital emergencies (heart attack, car accident…).

The pathology “general examination”: six rooms are dedicated for all the others pathologies.

We have used these different types of care rooms in order to specify the different categories of patients and we have taken into consideration different levels of gravity according to the pathology. The Figure 4 gives an example of a global representation with LAESH of the patient pathway “General examination”. This extract begins after the assessment by the emergency department and the complementary examinations which are done outside of the emergency department. This example represents nine paths and ten stages. OUT indicates the outside world (which contains both input and output of the system). The stages are numbered from top to bottom and from left to right from number 1. A patient can only take one path (the "or" between the different paths is exclusive). For example, if the patient takes the first path (stages 1 and 2), he is provided for by the adapted resources for gravity 2 and after, he leaves the emergency department. If he takes the second path (stages 1, 2 and 5), he is transferred to another department after the examination in the emergency department.

The detailed graphic representation describes, for each stage and for each category of patient, the sequence of elementary operations (the most precise level of an activity description) and the structure symbols: the stage beginning and end; the elementary service execution; the waiting delay; the passive resources seizing and releases; the loops; the processes which are completed in parallel; the stages and sub stages calls.



1
Return to the emergency department after complementary exams
2
Cares gravity 2
3
Cares gravity 3
4
Cares gravity 4
OUT
5
Transfer to another department
6
Transfer to the hospitalization unit of short duration
OUT
7
Transfer to another department
8
Transfer to the hospitalization unit of short duration
OUT
9
Transfer to another department
10
Transfer to the hospitalization unit of short duration
OUT
OUT
OUT
OUT
OUT
OUT

**Figure 4. Extract of the patient pathway 'General examination'**

The Figure 5 gives the detail of the stage 2 of the previous figure.

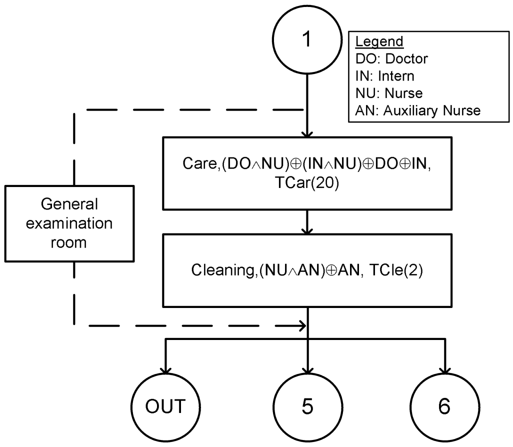


**Figure 5. Detailed description of the stage 2**

This stage begins after the stage 1 and it predates the exit, the stage 5 or the stage 6. This stage represents two elementary operations which use one passive resource (a general examination room) and the following active resources:

- The first elementary operation (care) uses a doctor with a nurse or an intern with a nurse or a doctor alone or an intern alone for the time "TCar(20)" (variable).
- The second elementary operation (cleaning) uses a nurse with an auxiliary nurse or an auxiliary nurse alone for the time "TCle(2)" (variable).

The knowledge model of the emergency department represents more than 140 paths and 190 stages with many identical stages (stage call). The different active resources are: doctor, intern, nurse, auxiliary nurse, hospital worker, orientation nurse and secretary.

### 4.3 Management rules

The management rules are detailed in this part which completes the knowledge model. These may be, for example, the conditions for the transition from a stage to one another (in the global representation) if they are not probabilities. At the detailed level, these management rules contain too: the priority rules, the pre-emption rules, the queue rules, the general organization rules.

For example, the main rules are:

- Priority and pre-emption: if there is a vital emergency (high gravity), the patient pre-empts the human resources (active resources) and the passive resources.
- Queue: the high gravity in first. For the same gravity, the rule is FIFO (first in, first out).
- General organization: during the day, a specific nurse is present for the welcome of the patients. During the night, this work is done by the others staffs.

## 5. THE TRANSITION RULES BETWEEN KNOWLEDGE MODEL AND ACTION MODELS

The transition rules between knowledge model and action models mainly concern the transition from the knowledge model, designed with UML and LAESH, to a simulation model. Regarding the transition from the knowledge model to action model carried out with Witness, Table 3 gives the transition rules. We created a Witness software library of components to match the requirements resulting from the specificities of the hospital systems.

**Table 3. Transition rules from the UML/LAESH knowledge model to the Witness action model**

| Knowledge Model | | Action Model |
|---|---|---|
| UML Object | LAESH | WITNESS |
| Patient | Category | Type of article |
| Care rooms | Passive resources<br>Seize, Release | Machine |
| Waiting rooms | Passive resources<br>Seize, Release | Stock |
| Human resources | Active resources | Operator |
| Activity | Elementary operation | Production cycle<br>Stock/Machine |

## 6. THE DECISION-MAKING AID TOOL

### 6.1 The input variables

The input variables of the decision-making aid tool are:

- The load of the system with quantity of patients expected by week according to the pathology and the gravity.
- The human resources schedules and the quantity of persons by type of human resource and by schedule.
- For each pathology and each gravity, the durations of elementary operations, the probabilities (supplementary examination, etc.) and the used human resources.

### 6.2 The observable variables

*For each human resource (or by type of human resource) on the week (or by day)*

Here, the main objectives being to specify the staffing requirements at the NHE, anticipating a possible increase in the workload of the emergency department, and to understand the origin of flow problems if necessary, we proposed the criteria:

- The occupancy time.
- The quantity of patients which are taken care of.
- For each type of human resources: the quantity of resources of each type used by comparison with the quantity of available resources on the week (by time interval of 15 minutes).

*For the patients*

In the hospital systems, the system performance is related too to the patient satisfaction. We have chosen to evaluate it by the following indicators which can be filtered by pathology, by day or by set of specific patients:

- The time spent in the care rooms.
- The time spent in the waiting room.
- The time spent at the supplementary exams (for example, in the radiology department for a fracture).
- The total time of the patient in the hospital.
- For each pathology, the mean representation of all these data with the percentage of each time (care, waiting time…) in the total time of the patients in the hospital.

*For the care rooms*

For information, we have decided to add representative indicators of the activity of the care rooms:

- The occupancy time.
- The quantity of patients which are taken care of.
- For each type of care room: the quantity of rooms of each type used by comparison with the quantity of rooms available on the week (by time interval of 5 minutes).

### 6.3 The user interface

The user interface enables the users to communicate with the simulation model. It has been created with Excel, using macros to automate tasks such as the arrival schedule of patients, or the generation of input files for the simulation model. This can create a friendly environment and interactive enough to be understood and used by non-specialists. In a fist time, the user interface enables them to parameterize and to customize all the input variables.

- The workload of the system: the interface can generate a patients arrival schedule based on the captured data and the chosen parameters (the quantity of patients, their distribution by pathology, by gravity level, by day and by time interval). This schedule gives the arrivals of the patient by time interval of 30 minutes on the week and it can be changed by the user before running the simulation.
- The human resources schedule: the users have to capture the different schedules and the quantity of assigned resources by schedule and type of resources.
- The different durations and parameters: the user can change for each elementary operation (activity) the duration and the used human resources. For each time (care, cleaning…), he can choose between a constant time and a probability law (uniform law for example). A statistical analysis on the field data is in prospect.

In a second time, after the launch of the simulation, the user interface enables to study all the results. As the results may be quickly readable by health care workers, we have chosen, for the main results, a graphical display. These results have been validated by the users.

In addition, after each simulation, a summary is automatically generated and presents the results, the input variables (with the patient arrivals and the resources schedules) and the objectives of the simulation (for example, a scenario with a supplementary human resource).

## 7. ANALYSIS

Following the data collection and the system modeling with LAESH, the input variables of the emergency model are captured with the information of the pediatric emergency department: quantity and distribution of the patients (forecasting of the NHE activity), the different durations, and the resources schedules. After the data entry, the simulation model is launched to test different organizational scenarios. The results of the simulation are collected and analyzed. Three scenarios are presented. The first scenario takes the quantity of hospital workers corresponding to the current organization of the pediatric emergency department and analyzes the consequences of that choice. The second scenario does not restrict the workers hospital and calculates the quantity of needed workers for the emergency department in order to reduce the waiting time for patients. The third scenario offers a quantity of workers hospitals possible for the pediatric ward of the NHE.

The data taken into account are those of the pediatric emergency department of the "Hôtel-Dieu" in order to use the simulation model on a real activity to assess the results. The physical structure and the management rules are those of the NHE.

### 7.1 The first scenario

The total time spent per patient in the emergencies department includes the time of care, the time of complementary examinations and the waiting time. One of the objectives of the pediatric emergencies is to reduce the waiting time. Figure 6 gives an example of the waiting time for the pathology 'Fracture'. To achieve this goal, two parameters could be modified: the quantity of rooms and the management of the hospital workers. The quantity of rooms being defined, the key parameter to analyze and modify the adequacy between the system workload and the available resources is the quantity and the schedules of the hospital workers. In view of the high quantity of patients in the evening which causes significant delays unlike the smaller quantity of patients in the morning, a paramedical hospital staff additional is needed. To add this staff, it is possible to create an additional post or to change a schedule in changing from day to evening. The simulation shows that if one or more medical hospital worker(s) held additional posts, they would contribute to reducing the waiting times. To define the necessary quantity of paramedical hospital workers to decrease the waiting times, it is necessary to define the maximum acceptable waiting time for the patients and increase resources.

### 7.2 The second scenario

To define the quantity of paramedical hospital workers, in order to decrease the waiting times, their quantity is not restricted. This scenario enables to obtain the smallest waiting time for the patients. The obtained results by the simulation give the quantity of resources of each type occupied by comparison with the quantity of resources available on the week. Compared with the first scenario, the second scenario gives a more precise quantity of hospital workers at any time of day throughout the week in order to reduce the waiting time for the patients. The results of the second scenario give the maximum used resource simultaneously: 2 doctors, 4 interns, 5 nurses and 5 auxiliary nurses. This scenario is an ideal and is not acceptable. One of our objectives is to propose a correct scenario (with a realistic quantity of resources and acceptable waiting times). Our approach consists in degrading the ideal solution (by decreasing the quantity of resources). The third scenario is an example with the results on an acceptable degradation of the ideal system.

### 7.3 The third scenario

To define the quantity of paramedical hospital workers needed to decrease the waiting times, their quantity was determined after the first and the second scenario. Compared with the first scenario, the third scenario gives on the overall results an improvement of the organization. For example, nurses are better employed in the third scenario as in the first scenario (Figure 7). It therefore represents a good compromise between the second scenario which is the ideal solution and the first one which is near of the current organization.

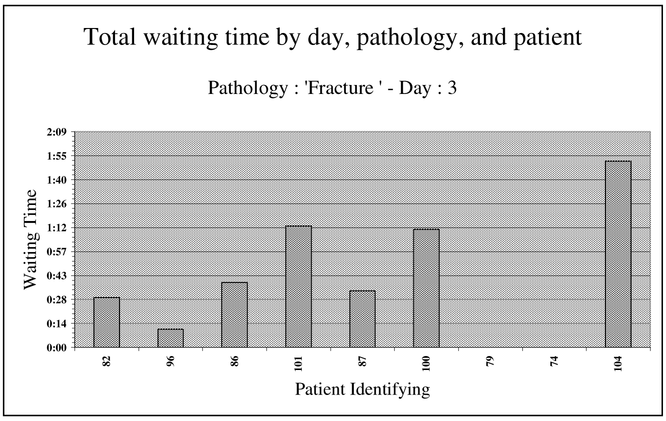


**Figure 6. Total waiting-time by patients and pathology**

### 7.4 Synthesis

In terms of methodology, the first scenario measure the impact of the current management rules on the structure of the NHE, the second scenario identifies the ideal organization in terms of staffing requirements and the third scenario is a compromise. The first scenario shows the current activity with some organizational difficulties and the second scenario presents an ideal organization with the same workload but by increasing the quantity of resources. The third one proposes to change the schedules of the interns, nurses, and auxiliary nurses.

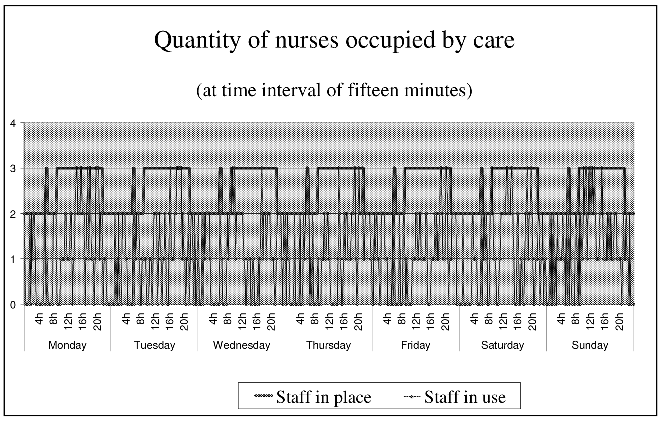


**Figure 7. Quantity of nurses occupied by care**

## 8. CONCLUSION

Today, hospitals have to minimize costs while maintaining a good level of performance for the patients and an acceptable quality of work to the nursing and medical staffs in all the departments. Our study deal with the emergency department which is a complex care unit in the sense that the organization of this unit differs from the other services by the quantity of treated patients with multiple heterogeneous diseases occurring at any time of day. We proposed in this paper, a state of the art work-related emergency department. The study of literature shows the dedicated problems, the lack of generics models and methodologies and the level of fineness which is sometimes little high for these models. Following the process modeling in ASDI, we are interested in a pediatric emergency department, and have designed a knowledge model of the emergency department. This model has been designed with UML and LAESH. We proposed a simulation model to analyze in detail several organizational scenarios. The simulation model has been developed with the Witness software. After filling the entry model, the simulation model is launched to test different organizational scenarios. Three scenarios have been analyzed. The results of this model are collected and analyzed. Our work enables us to propose a decision-making aid tool. This tool is now operational in the pediatric emergency department of the hospital. With this tool, operational decisions can be taken as the quantity of hospital workers present daily in the service. Some strategic decisions may also be applied as the choice of schedules of hospital workers over the year. This tool was validated by the hospital staffs and it can also be used to test exceptional situations as an unusual increase in the quantity of patients during a disaster. Finally, we plan to study the problems on various pediatric emergency departments to confirm our study. Further research concerns also optimization problems such as: resource assignment, planning …

## 9. REFERENCES

[1] Ambler, S.W. (2004) '*The Object Primer: Agile Model Driven Development with UML 2*', Cambridge University Press. ISBN 0-521-54018-6.

[2] Baesler, F.F., Jahnsen, H.E. and DaCosta, M. (2003) 'The use of simulation and design of experiments for estimating maximum capacity in an emergency room',

Winter Simulation Conference (WSC).

[3] Centeno, M. A., Giachetti, R., Linn, R. and Ismail, A. M. (2003) 'A simulation-ILP based tool for scheduling ER staff', *In WSC.*

[4] Chabrol, M., Sarramia, S. (2000) 'Object oriented methodology based on UML for urban traffic system modeling', *Lecture Notes in Computer Science*, vol. 1939, 425-439.

[5] Chabrol, M., Chauvet, J., Féniès, P., Gourgand, M. (2006) 'A methodology for process evaluation and activity based costing in health care supply chain', *Lecture Notes in Computer Sciences*, vol. 3812, 375-384.

[6] Chabrol M., Gourgand M., Rodier S. (2008) 'A modeling methodology and its application to the design of decision-making aid tools for the hospital systems', *In Proceeding of the International Conference on Research Challenges in Information*, 161-172.

[7] Chauvet, J. And Gourgand, M. (2008) 'Modélisation et analyse du service des urgences : état de l'art', *In conférence francophone en Gestion et Ingénierie de Systèmes Hospitaliers (GISEH)*, Lausanne.

[8] Faure, S., Dr Vermeulun, B. and Dr Wieser, P. (2003) 'Modélisation et réingénierie des systèmes hospitaliers', *In GISEH*, Lyon.

[9] Ferrin, D.M., Miller, M.J. and McBroom. (2007) 'Maximizing hospital financial impact and emergency department throughput with simulation', *In WSC.*

[10] Glassey, O. and Chappelet, J.L., (2002) '*Comparaison de trois techniques de modélisation de processus: ADONIS, OSSAD et UML'*, Working paper of l'IDHEAP 14/2002

[11] Gourgand M., Kellert P., (1992) 'An object-oriented methodology for manufacturing system modeling', *Summer Computer Simulation Conference*, 1123-1128, Reno (US), 27-30 Juillet 1992.

[12] Gunal, M.M. and Pidd, M. (2006) 'Understanding accident and emergency department performance using simulation'. *In WSC.*

[13] Huanxin, H.X, Mengchu, Z. and Monikopulos, C.N. (1994) 'Modeling and performance analysis of medical services systems using Petri nets', *IEEE International Conference on Systems, Man and Cybernetics*, vol.3, 2339-2342

[14] Hadges, P., Bellou, A., Grandhaye, J-P and Bayad, M. (2003) 'Modélisation de la prise en charge des patients du service des urgences', *In GISEH*, Lyon.

[15] Hay, A.M., Valentin, E.C. and Bijlsma, R.A. (2006) 'Modeling emergency care in hospitals: a paradox-the patient should not drive the process', *In WSC.*

*[16]* Jurishica, C.J. (2005) 'Emergency department simulations : medicine for building effective models', *In WSC.*

*[17]* Kolb, E.M.W., Lee, T. and Peck, J. (2007) 'Effect of coupling between emergency department and inpatient unit on the overcrowding in emergency department', *In WSC.*

[18] Komashie, A. and Mousavi, A. (2005) 'Modeling emergeny departements using event simulation techniques'. *In WSC.*

[19] Lubicz, M. and Mielczarek, B. (1987) 'Simulation modeling of emergency medical services', *European Journal of Operational Research*, vol. 29 (2), 178-185.

[20] Miller, M. J. and Ferrin, D. M. and Szymanski, J. M. (2003) 'Simulating Six-Sigma improvement ideas for a hospital emergency department', *In WSC.*

[21] Miller, M.J., Ferrin, D.M. and Messer, M.G. (2004) 'Fixing the emergency department: a transformational journey with Edsim', *In WSC.*

[22] Miller, M., Ferrin, D., Ashby, M., Flynn, T. and Shahi, N. (2007) 'Merging six emergency departments into one: a simulation approach', *In WSC.*

[23] Moreno, L., Aguilar, R. M., Piñeiro, J. D., Estévez, J. I., Sigut, J. F. and González, C. (1996) 'Using KADS methodology in a simulation assisted knowledge based system: application to hospital management', *Expert Systems With Applications*, vol. 10 (1), 17-27

[24] Navas, J. F., Arteta, C., Hadjes, P. S. and Jiménez, F. (2004) 'Construction et simulation d'un modèle de flux de patients dans le service d'urgences d'un hôpital colombien', *In GISEH*, Mons.

[25] Ruohonen, T., Neittaanmäki, P. and Teittinen, J. (2006) 'Simulation model for improving the operation of the emergency department of special health care', *In WSC.*

[26] Samaha, S. and Armel, W.S. (2003) 'The use of simulation to reduce the length of stay in an emergency department', *In WSC.*

[27] Siddharthan, K., Jones, W. J and Johnson, J. A. (1996) 'A priority queuing model to reduce waiting times in emergency care', *International Journal of Health Care Quality Assurance*, vol. 9 (5), 10-16.

[28] Sinreich, D. and Marmor, Y. N. (2004) 'A simple and intuitive simulation tool for analysing emergency department operations', *In WSC.*

[29] Sua, S. and Chung-Liang, S. (2003) 'Modeling an emergency medical services system using computer simulation', *International Journal of Medical Informatics*, 57-72

[30] Takakuwa, S. and Shiozaki, H. (2004) 'Functional analysis for operating emergency department of a general hospital', *In WSC.*

[31] Topaloglu, S. (2006) 'A multi objective programming model for scheduling emergency medicine residents', *Computers & Industrial Engineering*, 375–388

[32] Valenzuela, DT, Goldberg, J., Keeley, TK and Criss, A E (1990) 'Computer modeling of emergency medical system performance', *Annals of Emergency Medicine*, vol. 19 (8), 898-901

[33] Velin, P., Alamir, H., Babe, P., Four, R. and Guida, A. (2001) '*Les horaires principaux du circuit d'un enfant aux urgences pédiatriques'*, Expérience de l'hôpital Lenval en 1999, Archives pédiatriques.

[34] Wiinamaki, A. and Dronzek, R. (2003) 'Using simulation in the architectural concept phase of an emergency department design', *In WSC.*

[35] Yeh, J.Y. and Lin, W.S. (2007) 'Using simulation technique and genetic algorithm to improve the quality care of a hospital emergency department', *Expert Systems with Applications*, 1073-1083

# Interoperability and Healthcare

Miguel Miranda, Júlio Duarte, António Abelha, José Machado and José Neves
Universidade do Minho, CCTC, Departamento de Informática, Braga, Portugal
{miranda,jduarte,abelha,jmac,jneves}@di.uminho.pt

## ABSTRACT

In our research group work is being conducted on how to develop self organised Engineered Information Systems (EIS), which could enable researchers to create programs with significantly improved functionalities, leading to a more efficient and faster computation. These EIS are based on an original DNA origami which may be designed to serve as a scaffold for electronic workers (or software agents), going beyond existing technology either in terms of Socialization, Interoperability or the process of quantification of the Quality-of-Information (QoI) being exploited.

## KEYWORDS

Healthcare, Interoperability, Integration, Engineered Information Systems, Quality-of-Information, Multi-agent Systems

## INTRODUCTION

Many healthcare providers, designers, and practitioners in the field have questioned the relationship between people and care providers in the environment and sought empirical reasoning for the various guidelines in healthcare settings. Who'd believe it or not, an healthcare environment is based on an amalgam of diverse types of care provided by different departments and services that make the healthcare unit, where, without question, distinct communication and cooperation among them is essential.

On the other hand, with the introduction of new medical and information technologies, as well as novel methodologies for problem solving, the problem of interoperability turns into a rather intricate unsettled question. Furthermore, the use of point to point interoperation, connecting immiscible micro-computational environments available at their different departments and services, results in a rather complex web of interconnected communication channels which are difficult to monitor and hardly scalable [1]. These entanglements are a burden on the institution, but it becomes even more relevant when one moves to the process of interoperation among different institutions, where Business to Business integration (B2B) becomes essential to its proper functioning (i.e. secure sharing of the patients clinical information among different platforms). Indeed, these procedures are deeply impaired by bureaucratic, complex and paper based workflows, that new information systems and technological improvements aim to overcome.

Indeed, the problem of interoperability is becoming a cause for concern. The number of healthcare units being in use and the growth in the number of patients in need of care is increasing dramatically. A lot of factors may lead to medical errors. They may be, for instance, effects of medication, seasonal variation of weather conditions, or even human errors. From another standpoint, independently on the conditions under which a problem may occur, an operative response is needed.

Therefore, and in order to fulfill this goal, the research presented in this paper was directed to endorse the problem of medical error response, proposing an approach to a self-organization of resources available, in terms of the Heath Medical Record (HMR), under which it is publicized the current state of the patient and the way to be followed to overcome it. Such an environment is made up of a lot of heterogeneous resources as electronic devices, computational modules, computer-aided information sources, and human beings. The final purpose of the resource self-organization is to have their joint actions to response to a medical error. The resource functionalities are modeled by web services. For web service interactions, a formal interface agreement defined by the web services is used. To provide the web services with semantics, the web service descriptions are aligned against an ontology, which in this work is the SNOMED one[1].

In fact, this paper extends a previous one with regard to the issues of harmonization of web service descriptions and the ontology, as well as the service-oriented architecture of the medical error response system [9].

The use of common medical data standardized structures such as Health Level 7 (HL7[2]) and the dissemination of web services (WS) has, at different levels, great impact on enabling reliable methodologies of data exchange between software providers implemented at the several departments of the healthcare institution. However, the procedures leading towards the establishment of the interoperation process, such as the selection of used message types in HL7 or defining invocation arguments of web services, results in a dependance of continued support from the service providers. In other words, any healthcare institution that exists on numer-

[1] http://www.ihtsdo.org/
[2] http://www.hl7.org/

ous and diverse information systems, it becomes dependent of existing service providers and turns its computational infrastructure into a complex maze of possible fail-points.

On the other hand, the inherent distributed properties of agent oriented architectures present an excellent technology to overcome the interoperability problems addressed above, leading to agile, distributed and intelligent healthcare information systems [8]. Under these terms an Agency for the Integration, Archive and Diffusion of Medical Information (AIDA®[3]) was developed and implemented. This platform uses an agent oriented architecture to enable interoperation among applications of different service providers, while building a centralized datawarehouse.

Undeniably, a continuous demand for improvement is a cornerstone in order to provide better healthcare related services, either in terms of the discovery of new practices or techniques in the clinical realm, or the development and implementation of better information systems, that may ensure and promote the Quality-of-Information (QoI) gathered by these clinical breakthroughs. Following this line of thought, this paper also proposes new methodologies and technologies for problem solving that will be applied in the AIDA® platform.

## AN AGENCY FOR THE INTEGRATION, DIFFUSION AND ARCHIVE OF MEDICAL INFORMATION

The intricate distribution of information and services within a healthcare institution requires an integration platform to allow for the accessibility to all needed knowledge under different perspectives and intentions, following physicians, nursing, management and administration needs. With this intent, the Agency for Integration, Archive and Diffusion of Medical Information (AIDA®) was developed, using multi-agent systems for managing and storing the information in datawarehouses [9]. These agents store and organize the information, according to the specifications induced from the institutions architecture and needs, forming an ubiquitous network around an integrated global middleware and accepting specific requests and data from heterogeneous sources.

## THE COMPUTATIONAL MODEL

With respect to the computational model, it was considered extended logic programs with two kinds of negation, classical negation, ¬, and default negation, *not*. Intuitively, *not p* is true whenever there is no reason to believe $p$ (close world assumption), whereas $\neg p$ requires a proof of the negated literal. An extended logic program (program, for short) is a finite collection of rules and integrity constraints, standing for all their ground instances, and is given in the form [3]:

$$p \leftarrow p_1 \wedge \ldots \wedge p_n \wedge \text{ not } q_1 \wedge \ldots \wedge \text{ not } q_m; \text{ and}$$

$$?p_1 \wedge \ldots \wedge p_n \wedge \text{ not } q_1 \wedge \ldots \wedge \text{ not } q_m, (n, m \geq 0)$$

where ? is a domain atom denoting falsity, the $p_i$, $q_j$, and $p$ are classical ground literals, i.e., either positive atoms or atoms preceded by the classical negation sign ¬ [11]. Every program is associated with a set of abducibles [7]. Abducibles may be seen as hypotheses that provide possible solutions or explanations of given queries, being presented here in the form of exceptions to the extensions of the predicates that make the program. These extended logic programs or theories stand for the population of candidate solutions to model the universe of discourse.

Indeed, in our approach, we will not get a solution to a particular problem, but rather a logic representation (or program) of the universe of discourse to be optimized. On the other hand, logic programming enables an evolving program to predict in advance its possible future states and to make a preference. This computational paradigm is particularly advantageous since it can be used to predict a program evolution employing the methodologies for problem solving that benefit from abducibles [12], in order to make and preserve abductive hypotheses. It is on the preservation of the abductive hypotheses that our approach will be based, leading to a solution to the problem of healthcare demand and QoI, i.e., to model the universe of discourse in a changing environment, the breeding and executable computer programs will be ordered in terms of the QoI that stems out of them, when subject to a process of conceptual blending [12] [2].

Indeed, in blending, the structure or extension of two or more predicates is projected to a separate blended space, which inherits a partial structure from the inputs, and has an emergent structure of its own. Meaning is not compositional in the usual sense, and blending operates to produce understandings of composite functions or predicates, the conceptual domain. In other words, a conceptual domain has a basic structure of entities and relations at a high level of generality (e.g., the conceptual domain for journey has roles for traveller, path, origin, and destination).

Designing such a regime presents, still, unique challenges. Most evolutionary computation problems are well defined, and quantitative comparisons of performance among the competing individuals are straightforward. By contrast, in selecting an abstract and general logical representation or program, performance metrics are clearly more difficult to devise. Individuals (i.e., programs) must be tested on their ability to adapt to a changing environment, to make deductions and draw

[3] http://gia1.di.uminho.pt/aida (in Portuguese)

inferences, and to choose the most appropriate course of action from a wide range of alternatives. Above all they must learn how to do these things on their own, not by implementing specific instructions given to them by a programmer, but by continuously responding to positive and negative environmental feedback.

In order to accomplish such goal, i.e., to model the universe of discourse in a changing environment, the breeding and executable computer programs will be ordered in terms of the QoI that stems out of them, when subject to a process of conceptual blending. Therefore, let $i$ $(i \in 1, \cdots, m)$ denote the predicates whose extensions make an extended logic program that model the universe of discourse, and $j$ $(j \in 1, \cdots, n)$ the attributes for those predicates. Let $x_j \in [min_j, max_j]$ be a value for attribute $j$. To each predicate it is also associated a scoring function $V_{ij}[min_j, max_j] \rightarrow 0 \cdots 1$, that given the score predicate $i$, assigns to attribute $j$ a value in the range of its acceptable values, i.e., its domain. For the sake of simplicity, scores are kept in the interval $[0 \cdots 1]$, here given in the form:

*all(attribute-exception-list, sub-expression, invariants)*

This states that *sub-expression* should hold for each combination of the exceptions of the extensions of the predicates that denote the attributes in the *attribute-exception-list* and are according to the *invariants*. This is further translated by introducing three new predicates. The first predicate creates a list of all possible exception combinations (e.g., pairs, triples) as a list of sets determined by the domain size. The second predicate recurses through this list, and makes a call to the third predicate for each exception combination. The third predicate denotes sub-expression, given for each predicate, as a result, the respective score function. The QoI with respect to a generic predicate $K$ is, therefore, given by $QoI_K = 1/Card$, where $Card$ denotes the cardinality of the exception set for $K$, if the exception set is not disjoint. If the exception set is disjoint, the QoI is given by:

$$Q_k = \frac{1}{C_1^{Card} + \cdots + C_{Card}^{Card}}$$

where $C_{Card}^{Card}$ is a card-combination subset, with $Card$ elements.

The next element of the model to be considered, it is the relative importance that a predicate assigns to each of its attributes under observation, $w_{ij}$, which stands for the relevance of attribute $j$ for predicate $i$ (it is also assumed that the weights of all predicates are normalized [6]:

$$\forall i \sum_{j=1}^{n} w_{ij} = 1$$

It is now possible to define a predicate scoring function,

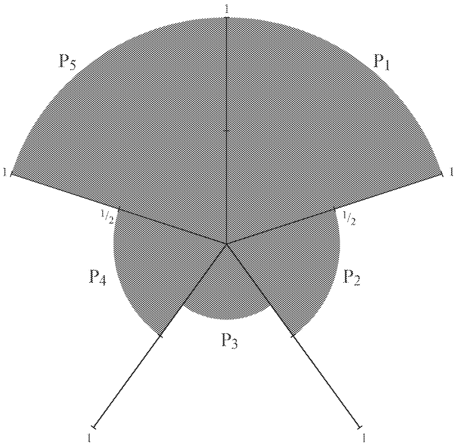


Figure 1: A measure of the QoI for logic program or theory P

i.e., for a value $x = (x_1, \cdots, n)$ in the multi dimensional space defined by the attributes domains, which is given in the form:

$$V_i(x) = \sum_{j=1}^{n} w_{ij} * V_{ij}(x_j).$$

It is now possible to measure the QoI that stems from a logic program, by posting the $V_i(x)$ values into a multi-dimensional space and projecting it onto a two dimensional one. Under this procedure, it is defined a circle, as the one given in Figure 1. Here, the dashed n-parts of the circle (in this case built on the extensions of 5 (five) predicates, named as $p_1 \cdots p_5$) denote the QoI that is associated with each of the predicate extensions that make the logic program P. It works out the most promising extended logic programs or theories to model the universe of discourse of the agents, providing the optimal solution, subject to formal proof, to the problem of healthcare demand and QoI.

It is also possible to return to the case referred to above, where we had a series of data that is produced according to a set of patient attributes, being got all time along. It is therefore possible, to produce a case memory, as the one depicted below, in terms of the predicates $itch$, $fever$ and $pain$ [4]. The corresponding evolutionary logic programs are presented in Figures 2, 3 and 4.

*The extended logic program for predicate itch*
$\{$
$\neg itch(X, Y) \leftarrow$ not $itch(X, Y) \wedge$
not $exception_{itch}(X, Y)$,
$exception_{itch}(X, Y) \leftarrow itch(X, itch)$,
$itch(john, itch)$,
$itch(carol, 1)$,
$exception_{itch}(kevin, 0.6)$,
$exception_{itch}(kevin, 0.8)$,
$?((exception_{itch}(X, Y) \vee exception_{itch}(X, Y)) \wedge$

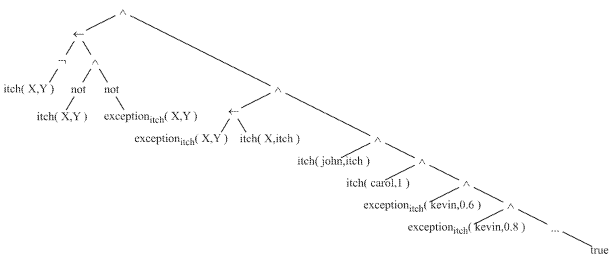


Figure 2: The evolutionary logic program for predicate itch

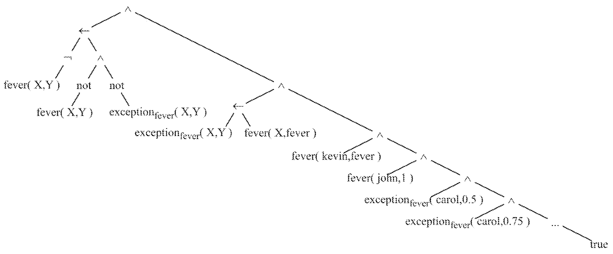


Figure 3: The evolutionary logic program for predicate fever

$\neg(exception_{itch}(X, Y) \wedge exception_{itch}(X, Y))$
$\}ag_{itch}$

*The extended logic program for predicate* $fever$
$\{$
$\neg fever(X, Y) \leftarrow$ not $fever(X, Y) \wedge$
not $exception_{fever}(X, Y)$,
$exception_{fever}(X, Y) \leftarrow fever(X, fever)$,
$fever(kevin, fever)$,
$fever(john, 1)$,
$exception_{fever}(carol, 0.5)$,
$exception_{fever}(carol, 0.75)$,
$?((exception_{fever}(X, Y) \vee exception_{fever}(X, Y)) \wedge$
$\neg(exception_{fever}(X, Y) \wedge exceptionfever(X, Y))$
$\}ag_{fever}$

*The extended logic program for predicate pain*
$\{ \neg pain(X, Y) \leftarrow$ not $pain(X, Y) \wedge$
not $exception_{pain}(X, Y)$,
$exception_{pain}(X, Y) \leftarrow pain(X, pain)$,
$pain(carol, pain)$,
$pain(kevin, 1)$,
$exception_{pain}(john, 0.3)$,
$exception_{pain}(john, 0.45)$,
$?((exception_{pain}(X, Y) \vee exception_{pain}(X, Y)) \wedge$
$\neg(exception_{pain}(X, Y) \wedge exception_{pain}(X, Y))$,
$\}ag_{pain}$

Now, and in order to find the relationships among the extensions of these predicates, we will evaluate the relevance of the QoI, which, for patient $kevin$, will be given in the form $V_{itch}(kevin) = 0.785$; $V_{fever}(kevin) = 0$;

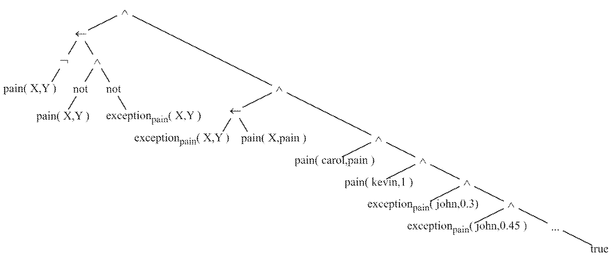


Figure 4: The evolutionary logic program for predicate pain

$V_{pain}(kevin) = 1$, i.e., it is now possible to measure the QoI that flows out of the logic programs referred to above (the dashed n-parts (here n is equal to 3 (three)) of the circles denote the QoI for predicates $itch$, $fever$ and $pain$).
It is also possible, considering what it is illustrated by Figures 5,6 and 7, to predict not only how the outcome of the patient diagnosis will fare into the future, but also how to fill in some missing data into the series. To do so, we need to evolve the logic theories or logic programs, evolving the correspondent evolutionary logic programs, according to the rules of programs synthesis [14] [12]. A new predicate may be defined (the three argument predicate $pathology$), whose extension may be given in the form:

$\{$
$\neg pathology(X, Y, Z) \leftarrow$ not $pathology(X, Y, Z) \wedge$
not $exception_{pathology}(X, Y, Z)$,
$pathology(john, flu, ((itch, 0), (fever, 1),$
$(pain, 0.785)))$,
$pathology(kevin, thrombosis, ((itch, 0.785), (fever, 0),$
$(pain, 1)))$,
$pathology(carol, heartattack, ((itch, 1), (fever, 0.785),$
$(pain, 0)))$,
$\}ag_{pathology}$

Now, given a new case, the seriation of the pathologies is made according the percentage of overlap between the dashed areas that make the QoI for the predicates in the case memory, and those for the new one. For instance, if we have a case under evaluation, with the QoI values depicted below:

$QoI_{itch} = 0.785$
$QoI_{fever} = 0$
$QoI_{pain} = 0.785$

it is possible to define an order relation with respect to the pathologies referred to in the case memory, leading to:

$Thrombosis > Flu > Heartattack$

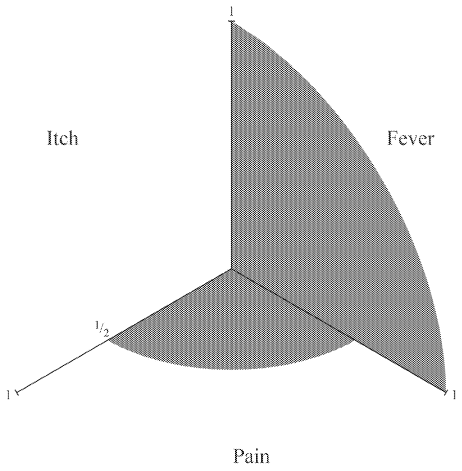


Figure 5: A measure of the symptoms for patient John

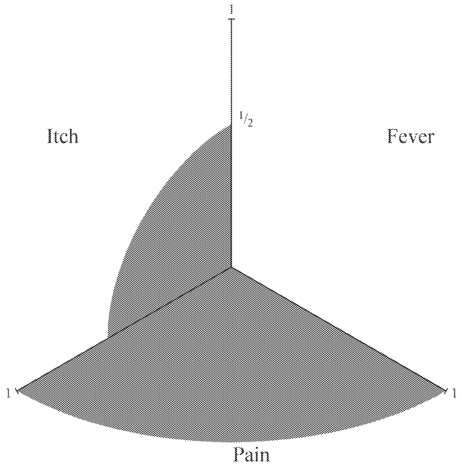


Figure 6: A measure of the symptoms for patient Kevin

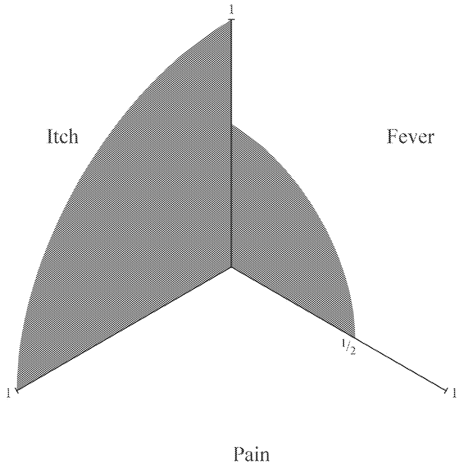


Figure 7: A measure of the symptoms for patient Carol

## AGENT ORIENTED MODEL

Indeed, the core units of the system are built as instances of independent and pro-active agents, which are in charge of problem solving tasks. Such a conglomerate of computational entities provide global back-office interoperation support, which may open the way to a consolidated data repository or warehouse storage systems. The analysis, specification and development of these systems, configures an outstanding field for the development of multi-agent architectures that may overcome the existing drawbacks, i.e., their disadvantages or problems that are being publicized.
Thereby, the platform agglomerates, structures and disseminates the information from different complementary diagnostic methods such as the radiological information system, as well as other existing ones. By this mean, balancing and fail-safe can be archived by agent migration among active platforms, while a clustered and mirrored repository can be used no unificate and standardized all patient's records and management indicators[10].
Architecture analysis demonstrate scalability is also not a problem as new agents can added and implemented without interfering with each other and new instances of the same agent base can be created to adapt to the systems needs on a a specific moment.
The development of a multi-agent system as a concept should always take in consideration some basic requirements for the platform:

- standardized communication and interactions;
- ability to manage agents locally and remotely;
- real-time global monitoring and dynamic contingency measures by the system;
- registry of available services and agents.

The usual concept of Agent Management System (AMS) and Service Registry Platform (SRP), is rather limited to large scale implementations, and therefore it is essential the existence of a global repository and management tier. Local platform's AMS and SRP, LAMS and LSRP respectively, are essential for the an hasten, distributed and independent architecture, ensuring the unavailability of the global AMS and SRP will not result on the collapse of the services they provide. Under this architecture agents of one platform will only comunicate with the LAMS and the LSRP, which in their turn interoperate with the global AMS and SRP, as is demonstrated in Figure 8.
Following this architecture, a local ASM is started in each platform which in its turn tries to sincronize with the global ASM in order to create the agents accordingly to these global definitions. If however unavailable, the LAMS will start the agents that are defined as vital in its configuration. By this mean the overall AMS rules

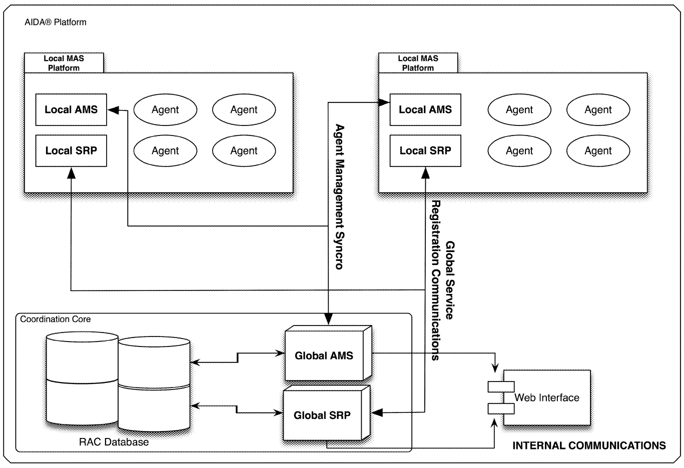


Figure 8: Agent Framework Architecture

extend over the local AMS rules, which will only be used on system failure.

Local instances can still manage their own agents and keep existing mapped services while the global AMS and SRP are unavailable. However, when the global AMS is available, the local AMS needs to synchronize the existing agents, their status and statistics to the global AMS, as well as to perform the management indications suggested by this platform. Nonetheless, the global AMS interacts with the global AMS, and provides an web interface in which local AMS and resulting agents are remotely and easily managed and monitored. Likewise, the global SRP allows for service distribution and dissemination on the overall system platforms, acting as constant yellow pages where the local SRP can display the services its agents provide and the services provided by other platforms. An diagram of message flow can be found in Figure 9. In this figure the flow for two different request of informations is demonstrated. The first request is a request based solely on agent communication, and in which a list of services is requested from the local SRP, which on its turn also requests and provides, if possible, a list of services performed by other platforms. On the other hand, the second request is not made by a software based agent, but rather by an human agent by means of an web interface. This interface is able to invoque an web-service, which is is provided by the global ASM. This process is performed on demand when possible or uses existing mapping on the local AMS if unavailable. The state of the agents and the platforms existing on the system (active or presently unavailable and already mapped) is stated in this request.

When a multi-agent system becomes vital for the functioning of an healthcare institution, the subject of agent monitoring and dynamic problem solving is essential to insure that the agent has not entered a state in which it is blocking the interoperation it is responsible for. For this reason all agents must report and provide regular state information to the LASM:

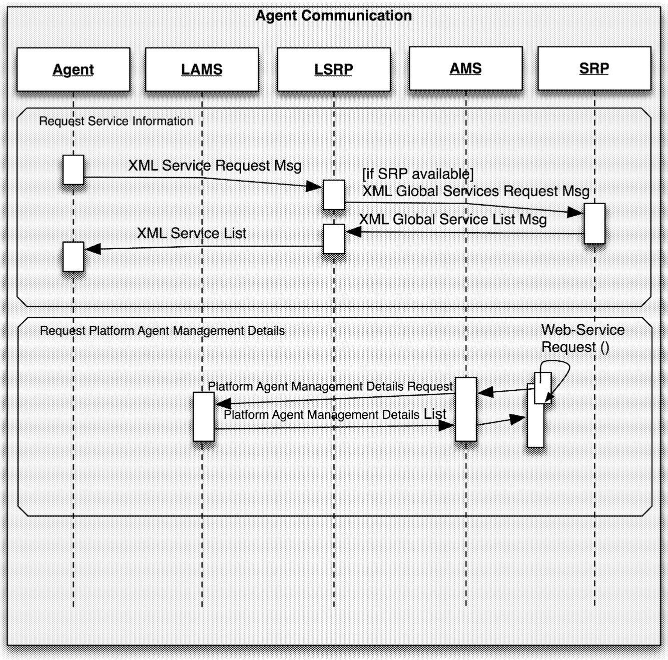

Agent Communication
Agent
LAMS
LSRP
AMS
SRP
Request Service Information
XML Service Request Msg
[if SRP available]
XML Global Services Request Msg
XML Global Service List Msg
XML Service List
Request Platform Agent Management Details
Web-Service Request ()
Platform Agent Management Details Request
Platform Agent Management Details List

Figure 9: Agent Communication

- presently served unique identified request (if exists);
- time frame of presently being served request or standby-time waiting for a request;

These data feeds allow the LAMS to extract important platform information for any given agent, according to a set of indicators:

- time frame since last contact with the LASM;
- average time frame since last contact with the LASM;
- average time of unitary processing - average time for processing a single regular interoperation request;
- maximum time frame of successful processing - the highest time spent to resolve a request;
- average time of inactivity - average time waiting for a request to be presented to the agent;
- maximum time frame of inactivity - the highest time spent waiting for requests;

In order for these indicators to have any concrete meaning, a learning time must be granted to the LASM, so that maximums and averages are adapted to the agent type at hand. Another essential indicator is the threshold considered for automatic action, being these defined by the diference of maximum and average plus half of the average. These indicators are shared with the global AMS so that the administration can be informed of the existing decision constrains and to allow the definition of global thresholds chosen not by LASM. This effort relates to the need of global knowledge of the indicators when agents migrate within platforms, as well as to improve the quality of existing local indicators through use of existing global indicators.

With the information extracted from these indicators the LAMS can assert when an agent has blocked in a state and at which request state problems occurred, being able to kill and create a new agent to process this request, as well as to produce warnings regarding this situation. Furthermore, when a provider fails to interoperate beyond the defined threshold, the LASM can also warn and react to this fact, resulting in pro-active monitoring and enabling the prediction of other providers, before it is actually reported by end-users.

These features add reliability, adaptability and scalability to the platform, further enabling interoperation and supervision of the multiple and heterogenous information system in the institution.

## HEALTHCARE ORIENTATED COMMUNICATION

As an healthcare oriented platform, agents are able to communicate either by HL7, structured XML and subsequently HL7 imbedded XML. HL7 is a constant in the arena this system is designed for, as the HL7 standard is a well defined and oriented protocol that, though slightly obscure to outsiders, had a great impact on health related systems [13]. This standard however is not meant for agents communication as does not encompasses the whole dominion of tasks and communications agent systems need to perform. An HL7 message, as the one exemplified on the Listing 1, is comprised of separated values depending of the type of message.

Listing 1: HL7 Message Example

```
MSH|^~\&|CLINIDATAXXI|MAXDATA|ALERT|MNI|
20090730001023||OML^O21^OML_O21|6265106|
P|2.4|1||||||||
PID|||PIDNBR^^^SONHO^NS||PATIENT^NAME^^^^^L||
19350209000000|F||||||||||9001078^^^SONHO
|||||||||||||||||||||
PV1||URG|||||||||||||||||9111184^^^SONHO
|||||||||||||||||||||||||||||||||V|
ORC|SC||354140#1^CLINIDATA||CM||||
20090730001023|80127^GERAL)^FATIMA
^^^^^^^SONHO|||||||||||||||
OBR|||354140#1^CLINIDATA|1^Am. URINA TIPO
II e SEDIMENTO^CLINIDATA|||||||O|||||
80127^GERAL)^FATIMA ^^^^^^^SONHO||||
512###URINA TIPO II c###125###
URINA BIOQUIMICA|||||F|||||||||||||||||||||||
```

The general usage of XML under this project relates to the extensibility of this technology and the importance of the structuration and adaptability it provides. By this mean not only the performative is considered but the context inherent to the XML structural definition. As exemplified on Listing 2 the message structure defines the performative and other essential information for the contextualization of the message in the XML tags that encapsulate the information.

Listing 2: Agent SRP service query

```
<MESSAGE performative=”request”>
  <INTERVINIENTS sender=”agentA”
     recipient=”agentB”>
  <INTERVINIENTS>
  <CONTEXT ontology=”AIDAservices”
     protocol=”AIDAv1.2”>
  </CONTEXT>
  <CONTENT type=”XML”>
    <AIDAREQSERV provider=”GE”
         query=”centricity” />
  </CONTENT>
</MESSAGE>
```

Although misusage of XML and abuse of its extensibility adds considerable overhead on communications and processing of messages, a planed, structured and aware XML based communication can be a competitive, adaptable and well dimensioned methodology, while maintaining all the qualities that define XML [5]. In the case of healthcare, specially in major health centers, the increased ease of interoperation based on the the usage of XML and the reliability it brings to the QoI storage and usage, overcomes the drawbacks previously described. On a further comparison of this XML solution and HL7, one may argue that XML allows the communication of existing information only, while HL7 contais the fields even if no information is contained.

## CONCLUSION

In this work it was presented a mathematical model that may significantly improve computers ability to automatically recognize and process data and knowledge, in terms of interoperability and integration. The symbolic processing algorithms used may eventually surpass the human beings brain mechanisms and help machines perceive and understand the world around them. Indeed, the mathematical model that was designed, mimics, in a highly simply manner, the neuronal processes that occur in an artificial brain at different time points. In terms of patient care, the system is able to predict the patient behaviour, and to act accordingly, i.e., from a neuroscientific perspective, the reactions of the model to stimulus from the environment seem much better to what is being observed with the human brain.

## REFERENCES

[1] Aier S. and Schonherr M., Evaluating Integration Architectures –A Scenario-Based Evaluation of In-

tegration Technologies. Trends in Enterprise Application Architecture, 2-14, 2006.

[2] Analide C., Abelha A., Machado J. and Neves J., An Agent Based Approach to the Selection Dilemma in CBR, in Studies in Computer Science, volume 162, Intelligent Distributed Computing, Systems and Applications,Badica, C., Mangioni, G., Carchiolo, V., Burdescu, D.D. (eds), Springer Berlin / Heidelberg, 2008.

[3] Analide, C., Novais, P., Machado, J., and Neves, J. Quality of Knowledge in Virtual Entities, in Encyclopedia of Communities of Practice in Information and Knowledge Management, Idea Group Inc., 436-442, 2006.

[4] Angeline, P. J. Parse Trees In: Evolutionary Computation 1: Basic Algorithms And Operators, T. Back, et. al. (Eds). Bristol: Institute of Physics Publishing, 2000.

[5] Ericsson M., The Efects of XML Compression on SOAP Performance, World Wide Web, 10, n. 3, 279–307, 2007.

[6] Jennings, N.R., Faratin, P., Johnson, M.J., Norman, T.J., O'Brien, and Wiegand, M.E. Agent-based business process management, In Journal of Cooperative Information Systems, 5(2-3):105-130, 1996.

[7] Kakas A., Kowalski R. and Toni F., The role of abduction in logic programming, in Handbook of logic in Artificial Intelligence and Logic Programming, Volume 5, D. Gabbay and C. Hogger and J. Robinson (eds), LNAI, Springer, Oxford University Press, 1998.

[8] Machado J., Alves V., Abelha A. and Neves J., Ambient Intelligence via Multiagent Systems in Medical arena; International Journal of Engineering Intelligent Systems, Special issue on Decision Support Systems; vol. 15, n.3, 2007.

[9] Machado J., Abelha A., Novais P., Neves J.C. and Neves J., Quality of Services in Healthcare Units, in Proceedings of the ESM 2008, Le Havre, France, 2008.

[10] Miranda M., Abelha A., Santos M., Machado M. and Neves J., A Group decision support system for staging of cancer, in Proceedings of the 1st International Conference on Electronic Healthcare in The 21st Century, City University, London, England, 2008.

[11] Neves, J. A Logic Interpreter to Handle Time and Negation in Logic Data Bases, in Proceedings of ACM 1984 Annual Conference, San Francisco, October 24-27, U.S.A., 1984.

[12] Neves, J., Machado, J., Analide, C., Abelha, A., and Brito, L. The Halt Condition in Genetic Programming, in Progress in Artificial Intelligence, EPIA 2007 (LNAI 4874), Guimarães, Portugal, 2007.

[13] Smith B. and Ceusters, HL7 RIM: An Incoherent Standard. In Studies in Health Technology and Informatics. IOS Press, vol. 124, 133–138, 2007.

[14] Teller, A. Evolving programmers: The co-evolution of intelligent recombination operators. In K. Kinnear and P. Angeline, editors, Advances in Genetic Programming 2. MIT, 1996.

# MODELLING AND SIMULATION OF THE STOMATOLOGY SERVICE FOR THE RMUHO

Khaled Belkadi
LAMOSI, University of Mohamed Boudiaf, USTO,
BP 1505 Oran M'Naouer, 31000 Oran, Algeria
E-mail: belkadi1999@yahoo.com, belkadi@isima.fr

Alain Tanguy
LIMOS, CNRS UMR 6158, University Blaise Pascal,
Les Cézeaux, 63173 Aubière Cedex, France
E-mail: tanguy@isima.fr

**KEYWORDS**
Hospital, Stomatology service, ARIS modelling, SIMULA and Flexsim model, Discrete-event simulation.

## ABSTRACT

Stomatology is the part of medicine that relates to the mouth and its diseases; originally practised by physicians, it was a standard medical speciality through the early 20th century but in the U.S. it is now the domain of dentists. Stomatology service is an essential part of the hospital. The aim of this paper is to model and simulate the Stomatology service of Regional Military and University Hospital of Oran (RMUHO) in Algeria using ASDI methodology. The Stomatology service is mainly composed of Dental Cares, Conservative Dentistry, Dento Facial Orthopaedics, Periodontology, Pathology and Dental Prosthesis.

Our aim is to study the utilization rates of rooms and doctors of the stomatology service. This will enable us to improve service performances. To achieve this goal, we first used the ARIS tool to specify the knowledge model and then we simulate models thanks to the SIMULA language and Flexsim to implement action models

## INTRODUCTION

Hospital systems are complex systems in which problems have to be solved such as the size and number of their critical resources, the enhancement of their efficiency or obviously the understanding of their operation. These problems concern performance evaluation they can be solved using modelling and simulation.

Modelling is a decision aid tool which prevents important financial investments. This paper describes the modelling and the simulation of the Stomatology service in the hospital system. The ASDI (Analysis, Specification, Design and Implementation) (Gourgand and Kellert 1991) modelling methodology is adapted and used for this system. It is based on the construction of two models classes: the knowledge model and the action models.

The aim of this work is to model and to simulate the Stomatology service of the Regional Military and University Hospital of Oran (RMUHO), following ASDI methodology, using ARIS tool (Architecture of Integrated Information System) (Sheer 2002) for the knowledge model and the SIMULA language and Flexsim for the action models.

The Stomatology service of RMUHO is mainly composed of six specialities: Dental Cares, CD Conservative Orthodontics, DFO Dental Facial Orthopaedics, Periodontology, Pathology and Prosthesis.

Our goal is to model and simulate the stomatology service in order to  study the utilization rates of rooms and doctors of this service.

## STOMATOLOGY SERVICE OF THE RMUHO

Regional military and University Hospital of Oran (RMUHO) has been in operation since January 2005 (Caimed 2007; Saihi 2006). This hospital complex is located at the southern outskirts of the city of Oran, on an area of 40 ha and includes the following:  Hospital, Heliport, Sport Complex, and a city of 300 apartments. The RMUHO in Oran incorporates several administrations and services (Belkadi 2007; Moussa 2009) among them there is stomatology service (Beladam and Belkadi 2008).

## MODELLING METHODOLOGY

The modelling methodology ASDI has been adapted to hospital systems (Gourgand and Combes 1994; Mebrek and Tanguy 2006; Mebrek, et al. 2007). The knowledge model describes the structure and the operating principle of the system in a natural or graphical language; it is built thanks to three subsystems (logical, physical and decisional). An action model is a translation of the knowledge model in a mathematical formalism or in a programming language enabling the evaluation of chosen performance criteria.

The main goal of the modelling methodology is to establish a knowledge model that is as generic as possible and that allows the execution of the action models specific to the systems of the domain. The knowledge model remains an open model which is enhanced by each domain systems study. The management of the knowledge and the execution of the action models imply the help of an open modelling environment in order to include new and more efficient methods and tools. The modelling environment (figure 1) eases information exchange between the project members and helps the conception of action models during the extraction of the information from the knowledge model. It's an attempt to introduce automatism in the modelling process with the formalisation of the knowledge, the analysis of the data to determine the characteristics of the system, the operational research and the simulation for the evaluation. Graphical representations and animation tools help verifying proper operation of the model. The first knowledge model of the hospital logical system has been formalised thanks to the ARIS tool that is appropriate to describe organisations, processes and activities (Green and Roseman 2000), as well as entity-relationship models (Chen 1976).

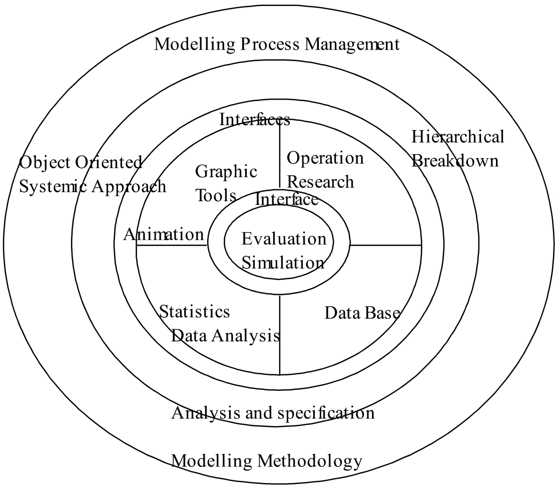


Figure 1: The modelling environment

## MODELLING PROCESS

Modelling is a set of techniques that provides the ability to study and understand the structure and the operating principle of a system. We use three rules to build a model that represents the reality: a model must be alike to the reality, a simplification of the reality and an ideal view of the reality. Figure 2 describes the modelling process.

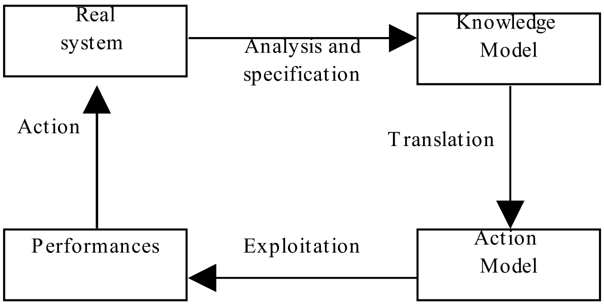


Figure 2: The modelling process of a system

The knowledge model is a formalised description of the system that contains the acquired knowledge during the observation phase of the existing system or the specification of the topology and functioning stated by the designers. The action model is a translation of the knowledge model using a mathematical formalism (for example an analytical method which takes advantage of the queuing network theory) or in a programming language (for example a simulation language). It is directly usable and states the performances of the modelled system without using direct measure. Exploitation of the knowledge model and of the action model is called modelling process. This process is generally iterative and consists in four steps which are the elaboration of a system knowledge model, the translation of this knowledge model into an action model, the exploitation of the action model to evaluate the performances of the system and the interpretation of the results and consequently to deduce the modifications to be made on the system. Each step includes a verification and validation phase. A knowledge model has a wide application area.

In order to use ARIS to design a knowledge model (Rob and Brabänder 2008; Güngöz 2004), several modelling hypothesis are to be taken into account:
- Each activity (function in ARIS) is linked to one or more organisational units of the hospital system (care unit, operating room, the pharmacy, the stomatology service, etc.);
- Each event possesses its own information document, it is used by several processes and it is referenced in one or more documents of the information system (medical file of the patient, file of the operating room suite, etc.);
- The referenced documents provide the knowledge concerning the key processes.

To match our modelling goals, we chose the ARIS tool-set and we retain two representation types (Sheer, 2000):
- The event-driven process chain (EPC) in order to show that the processes have a well defined structure and to control the logical subsystems flows;
- The organisational structure for the decisional subsystem to detail the relationships in and between the services.

## KNOWLEDGE MODEL EPC

The sequence of functions in the sense of an enterprise process is represented in process chain. In these chains, it is possible to indicate the departure and arrival events for each function. The events trigger functions and they are generated by them. Event-driven process chain (EPC) represents the organisational structure of the enterprise, i.e. the representation of the relationships between the data view, objects, the functions and the organisational views. An EPC describes the sequencing of functions. For each function, an initial event and a final event are defined. The events trigger the functions and a function generates events. Function and event are represented by a rounded rectangle and a hexagon (figure 3).

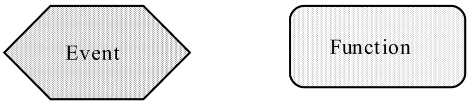


Figure 3: Function and event

As the events define the state or the condition that triggers a function as well as the end state, the start and end nodes of an EPC are always events. An event can trigger several functions simultaneously and a function can provoke several events. To represent the links and the processing loops of an EPC, the system uses a connector (or ruler) which as the shape of a circle. Figure 4 shows the specification of the Stomatology service by an EPC. Two connectors are used: an *And* operator and an *Exclusive Or*. The *And* operator insures that only an arriving patient having a recorded rendezvous may be processed. The *Exclusive Or* connector allows a patient to select only one speciality, this corresponding to the rendezvous.

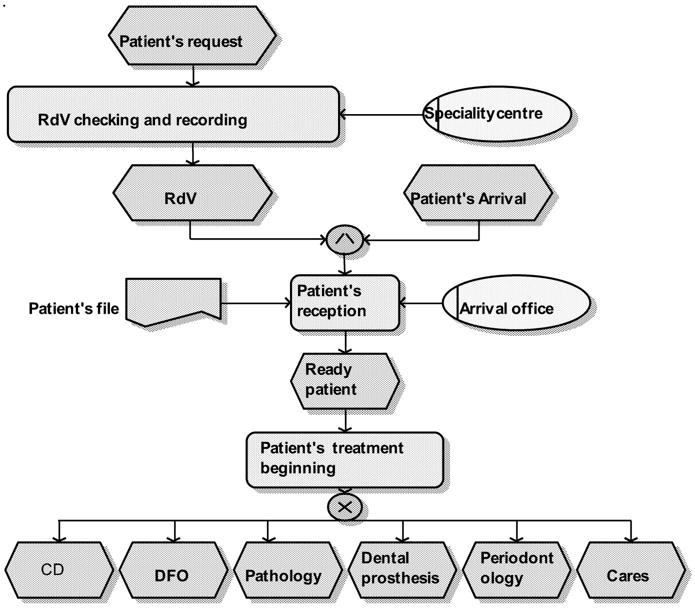


Figure 4: Stomatology service EPC

## ACTION MODELS

The Stomatology service action (or simulation) model of the "Regional military and University Hospital of Oran" (RMUHO) is represented by the SIMULA and Flexsim models.

### Queueing network model

The figure 5 shows how to use the waiting queue model to represent an action (or simulation) model of the Stomatology service. We model six units or sections: Dental Cares, CD (Conservative Dentistry), DFO (Dento Facial Orthopaedics), Periodontology, Pathology and Dental Prosthesis.

The system functioning is the following: a patient arrives, the secretary receives him and the patient waits for an available care room (in the waiting room). The service time of the secretary is a model parameter. Then he is prepared in the care room (preparation delay); the treatment begins when a doctor is available (treatment delay), at last he leaves the room, the room and the doctor are released.

The main specialities of the stomatology service are described by random delays characterized by their probability distributions. Some statistics do not appear in the scheme of figure 5 and depend on the evaluation tool. The patient arrival rates depend on the treatment or examination type and on the agenda (rendezvous times).

### SIMULA model

The SIMULA language has proved its capacity to implement different simulation models categories (Dahl and Nygaard 1965, 1966). It includes co-routines and processes of discrete events simulation. Numerous classes exist that extend the language possibilities regarding transactions management and statistical computations. The Gpsss class provides base objects such as the service, the storage, the transaction notions as well as the statistical region. Moreover, a simulation report is generated automatically. This class can therefore be used in GPSS programming with all the object-oriented capacity of a simulation language.

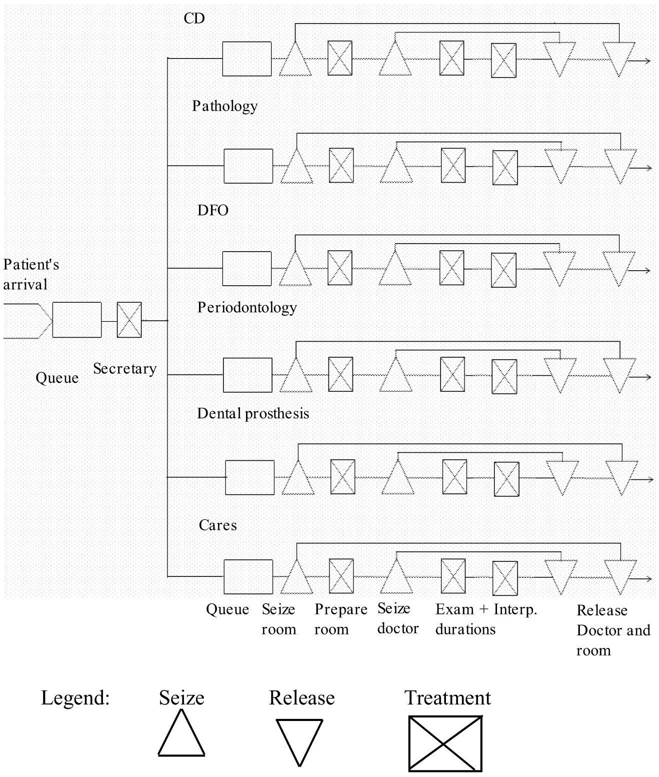


Figure 5: Queueing network model of Stomatology service

We realized a simulation based on an agenda (RdV rendezvous times) for one day. A patient obtains a rendezvous with one doctor for one care type at a given time. The RdV list is recorded in a Excel file. Obviously this list has a great impact on the utilization rates of rooms and doctors.

Table 1 shows a real case of RdV list of the RMUHO for 7 doctors (A to C and W to Z) and 6 specialities.

The reception activity of the secretary is modelled by a uniform probability distribution which duration is between 1 to 2 minutes. The secretary is represented by a *Facility* class of Montréal Gpsss external class.

The room preparation duration has been chosen constant 2 minutes. The treatment and reporting durations are regarded as uniform distributions which intervals depends on the speciality. The uniform distribution as very good properties such as simplicity, compact support and not too bad first approximation of more complex distributions. The table 2 describes the parameters of the models. The durations are given in minutes.

Table 1: Example of RdV list

| Doctor's name | Speciality | RdV time | Number of Patients |
|---|---|---|---|
| X | Pathology | 8.30 | 2 |
| Y | Periodontology | 8.30 | 1 |
| Z | DFO | 8.30 | 1 |
| X | Pathology | 8.45 | 1 |
| Z | DFO | 8.45 | 1 |
| Z | DFO | 9.00 | 1 |
| W | CD | 9.00 | 1 |
| W | CD | 9.15 | 2 |
| X | Pathology | 9.15 | 1 |
| Z | DFO | 9.15 | 1 |
| Z | DFO | 9.30 | 1 |
| W | CD | 9.30 | 1 |
| W | CD | 9.45 | 1 |
| A | Prosthesis | 9.45 | 1 |
| B | Prosthesis | 10.00 | 2 |
| X | Pathology | 10.00 | 1 |
| W | CD | 10.00 | 1 |
| W | CD | 10.15 | 2 |
| X | Pathology | 10.15 | 1 |
| C | Pathology | 10.15 | 2 |
| W | CD | 10.30 | 1 |
| C | Pathology | 11.00 | 3 |
| B | Prosthesis | 11.00 | 1 |
| W | CD | 13.00 | 3 |
| B | Prosthesis | 13.00 | 1 |
| D | Cares | 13.00 | 2 |
| Z | DFO | 13.00 | 2 |
| C | Pathology | 13.00 | 6 |
| C | Pathology | 14.30 | 4 |
| C | Pathology | 15.00 | 1 |

Table 2: Parameters of models

| | *Treatment duration (min - max)* | *Reporting duration (min - max)* | *Inter-arrival duration* |
|---|---|---|---|
| 1 CD (Odontics) | 15-30 | 3-5 | 15 |
| 2 DFO (Orthopaedics) | 20-25 | 4-6 | 20 |
| 3 Periodontology | 15-30 | 5-7 | 15 |
| 4 Pathology | 30-60 | 8-12 | 60 |
| 5 Prosthesis | 15-20 | 3-6 | 20 |
| 6 Cares | 20-30 | 1-2 | 20 |

## RESULTS AND INTERPRETATIONS

We obtained results with data taken on a real case for the Stomatology service of the RMUHO of Oran (Krour et al. 2008). Each speciality has a number according to table 2, that is used for rooms and pathology or examination types (figures 8 and 9).

The obtained results with models written in the SIMULA language and using Montréal Gpsss class are given in figures 6, 7 and 8. They are presented in 3 parts.

The first one gives a fragment of the simulation trace including the patient number, doctor's name, speciality type, event and time (figure 6).

The second part provides the performances of the resources (doctors and rooms) and statistical measures (durations of treatments for each speciality). The resources are modelled by *Storage* classes of Gpsss. For each doctor and room figure 7 gives mainly: the number of treated patients, the mean treatment times and the utilization rates.

```
1    X   Patho    Patient beg   30.000
2    Y   Period   Patient beg   30.000
3    X   Patho    Patient beg   30.000
4    Z   DFO      Patient beg   30.000
5    X   Patho    Patient beg   45.000
6    Z   DFO      Patient beg   45.000
2                 Patient end   55.000
7    Z   DFO      Patient beg   60.000
8    W   CD       Patient beg   60.000
4                 Patient end   61.000
1                 Patient end   72.000
9    W   CD       Patient beg   75.000
10   X   Patho    Patient beg   75.000
11   Z   DFO      Patient beg   75.000
12   W   CD       Patient beg   75.000
8                 Patient end   83.000
6                 Patient end   88.000
13   Z   DFO      Patient beg   90.000
14   W   CD       Patient beg   90.000
...
```

Figure 6: Beginning of the simulation trace

```
                     avg      avg.time     contents      util.
          entries  contents    transit    now    max       %
X             6      0.26       40.00      0      1     25.70
Y             1      0.02       22.00      0      1      2.36
Z             7      0.20       26.00      0      1     19.49
W            12      0.26       20.00      0      1     25.70
A             1      0.02       20.00      0      1      2.14
B             4      0.09       20.00      0      1      8.57
C            16      0.69       40.00      0      1     68.52
D             2      0.05       23.00      0      1      4.93
Room 1       12      0.27       21.00      0      1     26.98
Room 2        7      0.20       27.00      0      1     20.24
Room 3        1      0.03       23.00      0      1      2.46
Room 4       22      0.97       41.00      0      1     96.57
Room 5        5      0.11       21.00      0      1     11.24
Room 6        2      0.05       24.00      0      1      5.14
```

Figure 7: Ressources performances

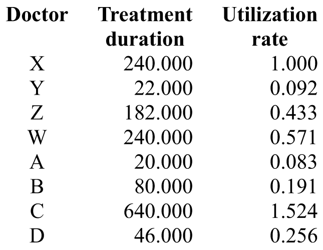
The statistics are automatically computed by means of *Region* classes of Gpsss. Figure 8 presents results by specialities and for the hospital: the number of processed patients, average contents and times including waiting duration of patients.

```
                      avg.      avg.time      contents      non-zero
            entries  contents    transit     now    max      transit
Hospital       49      7.85      149.63       0      19      149.63
 Patho 1       12      0.74       57.25       0       5       57.25
 Patho 2        7      0.39       52.00       0       3       52.00
 Patho 3        1      0.03       25.00       0       1       25.00
 Patho 4       22      6.46      274.09       0      13      274.09
 Patho 5        5      0.16       29.60       0       3       29.60
 Patho 6        2      0.08       39.00       0       2       39.00
```

Figure 8: Hospital and speciality performances

The third part provides, for each doctor, statistical results concerning treatment durations and utilization rates related to the working period of the doctor (figure 9).

The simulation starts at 0 and ends at a time that depends on simulation parameters and random number generator seed. For the given RdV list, the simulation stops at 933 minutes thus the last patient finishes his treatment 453 minutes later than the end of the working period. Three doctors are overloaded 12 patients exceed the normal load for doctor C and only one for doctor X and W. We may remark that the pathology treatment time is specifically high (30-60 minutes), in practice it is surely shorter or doctor C adapts its probability distribution to the known RdV list.



| Doctor | Treatment duration | Utilization rate |
|---|---|---|
| X | 240.000 | 1.000 |
| Y | 22.000 | 0.092 |
| Z | 182.000 | 0.433 |
| W | 240.000 | 0.571 |
| A | 20.000 | 0.083 |
| B | 80.000 | 0.191 |
| C | 640.000 | 1.524 |
| D | 46.000 | 0.256 |

Figure 9: Corrected utilization rates of doctors

The reception of the patients utilizes 5% of the secretary working time and the average through time is 1 minute for each patient. The utilization rates of the rooms are very different from 2.46% to 96.57%. It is the highest for room 4 which is the pathology room that clearly shows it is a critical resource. The average through time for pathology speciality is about 274 minutes with 6.5 patients in progress. This duration must be compared with the model parameters to conclude that waiting times are too high (treatment [30,60] minutes and report [8,12] minutes). When we take into account the working period of doctors and namely the pause between morning and afternoon we find doctor's utilization rates from 8.3% and 152%. Of course, the higher rate is obtained for pathology. It is due to three factors: there are too much patients (16 for doctor C and 8 for doctor X); the pathology room is a critical resource more rooms should be at disposal; the pathology rendezvous should be better distributed in a day and affected to the pathology doctors. We may although propose to increase the number of doctors or to make them able to shorten the treatment or examination time by means of new material.

The load of Cares (5.14%) and Periodontology (2.46%) is weak and it is not too high for CD (26.98%), DFO (20,24%) and Prosthesis (11.24%).

These results are obtained for one RdV list that represents only one day in a year so the study should be completed with the simulation of a lot of RdV lists and confidence intervals computation.

**Flexsim model**

For the same rendezvous list we have designed and realized a Flexsim (Flexsim 2009) model. According to the RdV list, the secretary receives the patients and dispatches each of them to one speciality service (CD, DFO, Pathology, Periodontology, Cares and Prosthesis). In each service rooms and doctors contributes to the treatment or examination of the patient. These resources are affected to the waiting patient when they are idle.

For the implementation of the Flexsim model we used available components such as *source* generating patients, *processor* of facility, *queue* for waiting, *operator* as resource and *sink* to free the generated patients. The instances of the model components are listed in table 3.

Table 3: Components of the Flexsim model

| Components | Identifiers | Comments |
|---|---|---|
| Source | Entry | Generator of patients |
| Queue | Waitsec<br>Waitexam | Secretary waiting room<br>Exam waiting room |
| Processor | Sec<br>CD<br>DFO<br>Perio<br>Patho<br>Prosthesis<br>Cares | Secretary<br>Conservative Dentistry<br>Dento Facial Orthopaedic<br>Periodontology<br>Pathology<br>Prosthesis<br>Cares |
| Operator | A<br>B<br>X<br>C | Doctor A<br>Doctor B<br>Doctor X<br>Doctor C |
| Sink | Exit | Treatment end |

The total treatment duration (processing) of each examination or care is given by the following formula:

Processing = Exam_duration + Reporting_duration

The distribution of processing durations are detailed in table 4. The setup duration of rooms is constant 2 minutes for the speciality of stomatology.

Table 4: Distributions of the processing durations

| Speciality | Exam duration | Reporting duration |
|---|---|---|
| OC | Uniform(15,30) | +Uniform(3,5) |
| ODF | Uniform(20,25) | +Uniform(4,6) |
| Periodontology | Uniform(15,30) | +Uniform(5,7) |
| Pathology | Uniform(30,60) | +Uniform(8,12) |
| Prosthesis | Uniform(15,20) | +Uniform(3,6) |
| Cares | Uniform(20,30) | +Uniform (1.2) |

The simulation starts at 0 and ends at 1267 minutes (about 20.37 hours). It is not incompatible with the results of the previous model. The computation of confidence intervals should confirm that. Table 5 presents the results provided by the Flexsim model.

Table 5: Results of the Flexsim model

| Resource room | Inp. | Avg. stay time | Content | | | State % | | |
|---|---|---|---|---|---|---|---|---|
| | | | Min | Max | Avg | idle | proc | setup |
| CD | 12 | 27.26 | 0 | 1 | 0.81 | 19.0 | 75.0 | 5.9 |
| DFO | 7 | 29.41 | 0 | 1 | 0.57 | 43.3 | 52.9 | 3.9 |
| Periodontology | 1 | 26.19 | 0 | 1 | 0.44 | 93.3 | 6.2 | 0.5 |
| Pathology | 22 | 56.18 | 0 | 1 | 0.98 | 2.5 | 94.1 | 3.5 |
| Prosthesis | 5 | 23.82 | 0 | 1 | 0.37 | 69.5 | 27.9 | 2.6 |
| Cares | 2 | 27.88 | 0 | 1 | 0.15 | 85.7 | 13.2 | 1.0 |

Table 6 presents the work durations of doctors and their utilization rates according to their working period: morning 240 minutes, afternoon 180 minutes and full day 420 minutes.

Flexsim automatically compute the work duration for each doctor. The utilization rates have been manually computed as the quotient of the work duration by the due working period.

Table 6: Results for the doctors

| Doctor | Working period minutes | Work duration minutes | Utilization rate |
|---|---|---|---|
| X | 240 | 309.8 | 1.290 |
| Y | 240 | 24.2 | 0.100 |
| Z | 420 | 191.8 | 0.456 |
| W | 420 | 303.2 | 0.721 |
| A | 240 | 87.4 | 0.364 |
| B | 420 | 21.7 | 0.051 |
| C | 420 | 882.1 | 2.100 |
| D | 180 | 51.8 | 0.287 |

We remark that the utilization rate of the Pathology room is the highest 94.1% of the simulation duration i.e. 247.8% of the working period. The Pathology doctors are overloaded: doctor C 210% and doctor X 129%. The rooms and doctors utilizations of Periodontology, Prosthesis and Cares specialities are little. They are medium for DFO and CD.

The results of the Flexsim model seem less optimist than those provided by the SIMULA model but they confirm the tendency that we have previously remarked. The computation of confidence intervals would confirm the equivalence of both models.

## CONCLUSION

We presented the Stomatology service of the RMUHO and the usefulness of the knowledge model specified thanks to the ARIS tool, as well as the transition from this model to the action (simulation) model implemented with the SIMULA language and Flexsim.

The SIMULA language and the Gpsss class is very suitable in modelling, simulation and statistics computing. Flexsim has interesting ability in quick development and animation but the automatically obtained results are not exactly those we need.

Both models have equivalent behaviours. A computation of confidence intervals would confirm they validate each other.

For one rendezvous list we discovered the overload of one speciality service. A wider statistical study of stomatology service should provide finest parameters. More real RdV lists should show the real behaviour of the service that is necessary to tune the dimension of the system and to optimize its functioning.

Several research themes can complete this work: modelling and simulation of other hospital services, using other modelling and simulation tools like Witness and animation tools, studying the planning of hospital services and the driving of the hospital systems.

## REFERENCES

Beladam, D., k. Belkadi 2008. "Service of Stomatology of HMRUO". Internal Report No. 3, Computer Science Department, USTO, Oran,

Belkadi, K. 2007. "Regional military and University Hospital of Oran (RMUHO)", Internal Report No. 1, Computer Science Department, USTO, Oran,

Caimed. 2004. "Social assistance in the Mediterranean Region Algeria ".

Chen, P. 1976. "The entity relationship model–Toward a unified view of data".*ACM Transaction on data base system*, Vol 1, N°1.

Dahl O. J. and K. Nygaard. 1965. *SIMULA - A Language for Programming and Description of Discrete Event Systems*. Introduction and User's Manual. Norwegian Computing Center, Oslo.

Dahl O. J. and K. Nygaard. 1966. "SIMULA - An Algol-based Simulation Language". *Communication of the ACM*,No. 9.

Flexsim Products Inc., 2009. "*Flexsim – Process simulation software*", http://www.flexsim.fr/solution1.html

Gourgand, M. and C. Combes. 1994. "*A modelling environment for hospital systems".* Ph.D., Clermont2 University, Clermont-Ferrand, France.

Gourgand, M. and P. Kellert. 1991. "Modelling environment design for manufacturing systems". *3ème congrès international de Génie Industriel*, Tours, France.

Green, P. and M. Roseman. 2000. "Modelling: an ontological evaluation". *In Information systems, ATED process*, Vol. 25.

Güngöz. Ö. 2004. "ARIS Architecture of Integrated Information Systems". Objectives lecture 2004, Aoyama Gakuin University AGU summer term Japan, Url: http://www.bpm-agu.com/downloads/Summary_ARIS.pdf

Krour, D.; D. Beladam; and K. Belkadi. 2008 "*Modelling and simulation of a hospital system: the case of ophthalmology and stomatology service of RMUHO -Oran*". Memory of computer engineer, USTO, Oran, Algeria.

Mebrek, F. and A. Tanguy. 2006. "Modelling and discrete events simulation of the imagery pole of a modern hospital". *6ème Conférence Francophone de Modélisation et Simulation - MOSIM'06,* 3-5 avril 2006, Rabat, Morocco.

Mebrek, F., k.; Belkadi; M. Gourgand; A. Tanguy. 2007. "Modelling and Simulation of the External Sterilization for New Hospital Estaing". *Colloque sur l'Optimisation et les Systèmes d'Information* (COSI'07) 11-13 juin, Oran, Algeria.

Moussa M. and K. Belkadi. 2009. "Simulation of flows in a service of the imaging RMUHO", 2nd International Conference SIIE-2009, Hammamet, Tunisie.

Rob D., E. Brabänder 2008. "ARIS Design Platform: Getting Started with BPM". Springer.

Saihi A., 2006. "The public health system in Algeria: Analysis and Prospects ". Gestion Hospitalière

Sheer, 2000. "Aris 5.0 Method", Edition IDS Scheer AG, Saarebruck.

Sheer. A.W. 2002. "*ARIS-Business Process Modelling*". Springer.

# DATA SIMULATION AND STORAGE

# Modelling of VoIP Overlay Routing using Graph Transformation

Ajab Khan and Muhammad Muzammal
Department of Computer Sciences
University of Leicester, UK
ak271,mm386@le.ac.uk

## Keywords

Modelling Skype, VoIP, Graph Transformation

## ABSTRACT

P2P VoIP traffic potentially suffers from Quality of Service (QoS) issues such as packet loss, jitter and echo. Packets loss and jitter are mostly caused by the reconfiguration in the overlay topology and peer dynamism in the P2P architecture. Peer dynamics and complexity of the P2P network make it hard and expensive to validate these solutions through testing and traditional simulation techniques. We are using Skype as case study and we address the issue of QoS by modeling network using graph transformation, and using stochastic analysis and simulation for validation.

## 1. INTRODUCTION

The use of internet has increased in the last few years, due to higher bandwidth availability, rapid advancement and development in the P2P applications. P2P Voice over the Internet Protocol (VoIP ) application are getting popularity amongst the internet surfers because of their low cost and easy access. P2P VoIP uses P2P to search another client and to relay voice packets [5]. P2P VoIP overlay networks have been the subject of interest for researchers recently [15, 13].

P2P systems are decentralised [3] self-organizing systems that are build on top of the physical network using overlay topology. Overlay topology can provide better performance by routing traffic through overlay nodes rather than using the commercial public routes advertised by the Internet Service Providers (ISP). Peers in P2P network are always in full control of their local resources and peers can change or impose new policy regarding their use in overlay networks [12]. Hence, overlay topology can not put any constraints on the peer arrival or departure. There is no central point of command and control. Due to lack of global control, peer dynamism and unreliability of the infrastructure, these systems are prone to dependability problems. This makes these systems different from other distributed systems such as client server model. Another important aspect of the P2P systems is that roles are dynamic and mostly emerging. It means that roles in the P2P architecture are not static and not pre-defined as in the client server model. P2P systems rely on emerging and dynamic roles as a result of an ongoing self-organization.

Since, data and voice frames flow through intermediate overlay nodes, but nodes are free to join and leave the network, the node may even impose a constraint to block routing traffic for third parties. This may result in need for topology reconfiguration and in case of VoIP the network has to recover fast enough so that Quality of Service (QoS) is not affected [2].

P2P VoIP traffic potentially suffers from QoS issues such as packet loss, jitter, echo and latency. Packet loss and latency are mostly due to overlay topology reconfigurations caused by peer dynamism or change of roles, whereas increased jitter is due to packets arriving at variable time intervals mainly because of network congestion, re-routing or peer dynamism.

There is some work on path selection in overlay networks and most of the work is for data packets rather then voice packets which have different latency and bandwidth requirements. [5] proposes autonomous system aware peer relay protocol called ASAP. The objective is to improve the QoS, system scalability with low overhead. [3] proposes to keep many redundant links between peers, so in case of link failure, alternate link is readily available. This approach does not seem feasible when there are millions of peers involved. [1] proposes an incentive based approrach where incentive should be given to all intermediate facilitating nodes and the resource owner. [4] proposes changes in the routing strategies. [16] uses an approach to select overlay path based on the available bandwidth measurements. Existing approaches are limited and peer dynamics and complexity of P2P network make it hard and expensive to validate theses solutions through testing and traditional simulations techinque [3, 6, 9].

We propose to model complex network reconfigurations. Our aim is to model different protocols in order to evaluate and improve their QoS properties with specific focus on VoIP applications. We consider P2P network architecture as a graph, the network nodes are represented as graph vertices and network connections are represented using graph edges. Reconfigurations based on route selection can be naturally modeled using graph transformations [3]. Further, stochastic analysis and simulation techniques could be used for validation [9].

In this paper, we present a case study of popular VoIP application Skype and discuss how to face some of the challenges posed by it.

## 2. BACKGROUND

Graphs represent the most basic model for entities and relations. Graphs are popular to be used for abstract representation of models. In UML model, a collection of object graphs is represented by means of a metamodel as abstract notation. Formally, the basic version of a graph consists of a set of vertices $V$ and a set of edges $E$ such that each edge $e$ in $E$ has a source and a target vertex $s(e)$ and $t(e)$ in $V$, respectively.

More, advanced version of graph models use attributed graphs [7] in which the nodes and edges can be associated with attributes carrying textual, boolean and numeric information. Graphs can also use the concept of inheritance in their node types. Graphs occur at two level: type level and instance level. The type level is used to represent the metamodel and instance level as object graphs. In simple words, graphs provide a modeling language, where graph model is *states* and *rules* provide state changing operation. The concept can be more generally described by a type graph $TG$ where the fixed type graph $TG$ represents all the possible states in the model and the instance graph represents individual snapshots, whereas the transformation of instance graph results from application of transformation rules.

A graph transformation rule $p : L \longrightarrow R$ consists of a pair of $TG-typed$ instance graphs $L$, $R$ such that the intersection $L \cap R$ is well defined. The left-hand side $L$ represents the pre-conditions of the rule where as the right-hand side $R$ describes the post-conditions or result. Their intersection represents the elements that are required, but not destroyed, by the transformation [3].

## 3. A GRAPH BASED MODEL FOR SKYPE

Skype is VoIP application developed by KaZaa in 2003. Currently, it has more than 170 million [17] registered users out of which 10% always remain online. Every day millions of people use skype to make free Skype to Skype calls or call PSTN numbers at cheap rates. Skype is also used to send files, instant messages and even for real time free vedio chat. Skype uses P2P architecture to exchange information among clients. The use of P2P and overlay architecture makes Skype unique as it uses the bandwidth of clients to maintain overlay network and serve clients. Since, it uses user's bandwidth therefore, it has been banned in universities and other organizations.

Skype nodes are distinguished into Skype client and Super node which is the encoded extension of the Skype client. Skype client based on the perceived network bandwidth, memory and network connectivity can promote to the role of the super node while retaining the primary properties of the Skype client. The super node form overlay network where a Skype client has to select one of the super node as host. The host super node will be used to search clients in the network and forward data or voice packets for the client. The super node acts as router or telephone switch in the network [8].

The architecture of Skype is described with help of a type graph in Figure-1. The type graph consist of Registration Server (RS), Super Node (SN-Ext), Skype Client (SC), a type Node generalises both SC and SN-Ext, a Packet Node, a Packet Node generalises both Call and Call Path. The edge types used are registration, link, overlay, route, sender, receiver, caller, callee, routepath and *at.*

In our model we assume that whenever a new SC is registered with Registration Server (RS), in the next step it has to select one of the SN-Ext as host. To select an SN-Ext the *SC* performs a latency test and if the round trip latency is less than or equal to standard latency value specified by International Telecommunication Union(ITU-T) [10], SC selects that *SN-Ext* as host SN-Ext and SC establishes *link* with the SN-Ext. In our model an SC with maximum bandwidth, memory and having static IP, is promoted to the new role of the SN-EXT. After promotion to the new role, SN-Ext becomes a part of the Overlay network established among the other SN-EXT.

In the event of communication, the caller SC sends the address of the callee to the host SN-Ext and the host SN-Ext finds the best path based on three parameters *Bandwidth* (B), *Latency* (L) and *No.of Hops* (H). The SN-Ext after receiving request from caller SC finds the best path based on some objective function and conveys the selected path to the caller SC for transmission of voice packets. The caller SC sends voice packets on the prescribed path and if the packet loss and jitter increases due to congestion, peer departure or change in the policy of the Overlay node, then callee SC sends message to the host SN-Ext of the caller SC. Upon receipt of this message, host SN-Ext changes the route for the remaining packets for the conversation. In the model, if during a call host SN-Ext of caller leaves the network or downgraded to the role of the client. The caller SC may select the next SN-Ext in the path of the call as new host of the SC.

## 4. SKYPE OVERLAY ROUTING AS GRAPH TRANSFORMATION

We now present a set of rules based on a scenario that we want to model. Suppose, a client is connected to the host super node and it decides to place a call to another SC. The Skype client (SC) sends a message to the host super node which contains the address of the callee client. The host super node sends time stamp packets on all possible routes to the target callee. Here, we are using the concept of the multi-objects in which multiple time stamped packets are created and are transmited on all out going edges. These packets record the bandwidth and hops while traveling from one super node to the other super node. The modeling of time follows the unique time stamp attributes chronos associated with node of the graph [9]. The chronos is used to keep track of the time from one event to other. The chronos attributes measure the time taken to complete a round trip. In order to stop packets circulating in a ring, we put constraint that if any packet has *hop* value greater than *15* then the packet shall be deleted. Since, deleted packet will be lost so that particular route will be considered in-feasible.

When these packets reach the callee client, the callee client returns all packets received to the host super node of the caller client. The host super node upon receipt of these packets calculates the round trip time, average bandwidth and hop. The host super node use an objective function to find the best path. The best path is communicated to the caller client and after that actual transmission starts

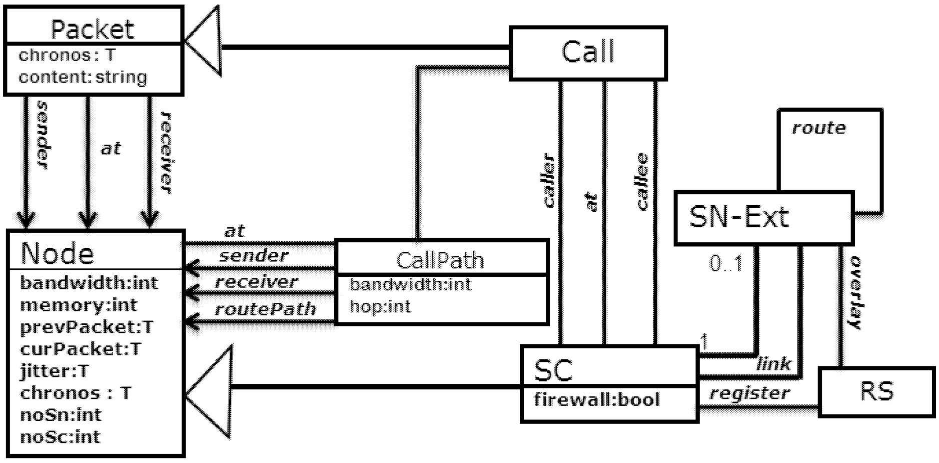


**Figure 1: Type graph**

using the best path provided. If during communication best path behaves to increase latency or increased packet loss the host super node upon receipt of message from the callee can switch the traffic to the next best path based on the calculation performed for this specific call.

**Rule in Figure 2: Place call**. This rule is used to inform the host SN-Ext that SC wants to place a VoIP call to another SC. Since, host SN-Ext is serving a number of SCs therefore, it keeps record of who is calling who. After the message is received, SN-Ext has to select the best path for the respective call. Here, we are using the concept of the negative application ondition (NAC) [3]. A *NAC* assures that the rule will only be applied if the specified element does not exist. It means if the client is not already in call then it can make a new call to another client.

**Rule in Figure 3: Create multiple packets and forward to super node**. This rule creates multiple packets and forwards them to the super nodes connected to current host SN-Ext. Creation of multiple packets follow the approch of the multi-objects creation. This rule also copies the current system time into the attribute *chronos* and initializes *bandwidth* and *hop*. The bandwidth and hop are updated as the packets travel from one super node to the other.

**Rule in Figure 4: Transfer of packet from one super node to other**. This rule forwards the packets from one super node to other along the path and updates corresponding attributes.

**Rule in Figure 5: Forwarding packets to callee skype client**. This rule forwards multiple packets towards a single callee client.

**Rule in Figure 6: Return all packets to host super node**. This rule returns all packets that were transmitted towards the callee client. The host super node finds the latency for each route by using $latency = (sn1.chronos - cp1.chronos)$ and the average bandwidth is computed using $bandwidth = ((cp1.bandwidth)/(hop))$. After computing the latency and bandwidth for each route, we use an objective function to find the best route that has maximum *bandwidth*, minimum *latency* and minimum *numberofhops* as per requirements of the QoS of the VoIP reflected by the ITU-T [10]. We aim to define an objective function $f$ that maximizes *Bandwidth* and minimizes $Latency(L)$ and $Number$ of $Hops(H)$. There are two issues that need to be addressed. First, the parameters under consideration namely $B$, $L$ and $H$ have different data ranges. For example, $B$ has a data range say from 56 $k$ to 20000 $k$ ($k$ stands for kilobytes), but for $L$ it is between 150 $ms$ and 300 $ms$ ($ms$ stands for millisecond). Second, there is a possibility that under given circumstances, one parameter say $B$ may have more significance than other say $H$. Obviously, $B$ should be given more weight than $H$ in such a case. We define an objective function $f(B, L, H)$ as follows: We assume that for parameters $B$, $L$ and $H$, we have some upper and lower bounds in the form of *min* and *max* values. First, we address the issue of different data ranges by normalizing the data values for all parameters. For a given data value $d$, we normalize it using following formula:

$$\delta = \frac{d-d_{min}}{d_{max}-d_{min}}$$

where $\delta$ is the normalized data value as $0 \leq \delta \leq 1$. $d$ is the original data value and $d_{min}$, $d_{max}$ are minimum and maximum data values accordingly. Next, we define an objective function $f(B, L, H)$ such that it maximizes $B$ and minimizes $L$ and $H$, and also gives more weight to more significant parameter(s):

$$f(B, L, H) = (a \times B^2) - (b \times L^2) - (c \times H^2) + (d \times B) + (e \times L) + (f \times H) + g$$

where $a, b, c$ are constant coefficients. $d, e, f$ are corresponding weights for $B$,$L$ and $H$ such that $\sum(d, e, f) = 1$. $g$ is also a constant. To check the effectiveness of our objective function $f$, we have used sample data in Table 1. We have assigned sample values to constants like for $a$,$b$ and $c$, we set the values as $'1'$. For $d$,$e$ and $f$ values are set as 0.60, 0.25 and 0.15 respectively. $g$ is set as $'2'$, so that we do not get -ive values for $f$. The values shown in bold depict the best possible route as it has maximum score for data under consideration.

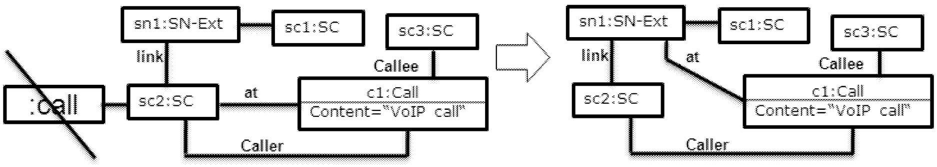


Figure 2: Place call

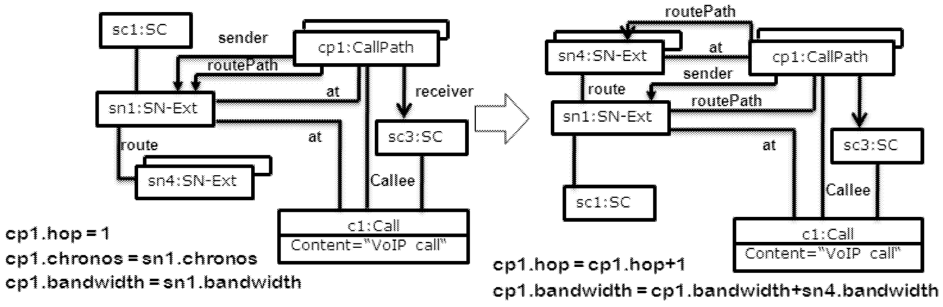


Figure 3: Create multiple packets and forward to super node

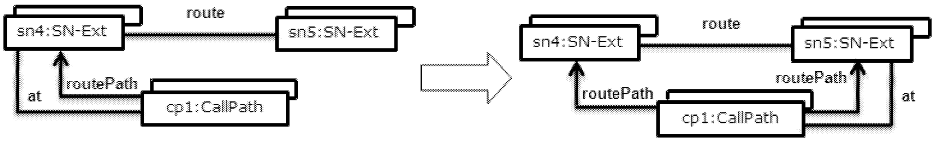


Figure 4: Transfer of packet from super node to other

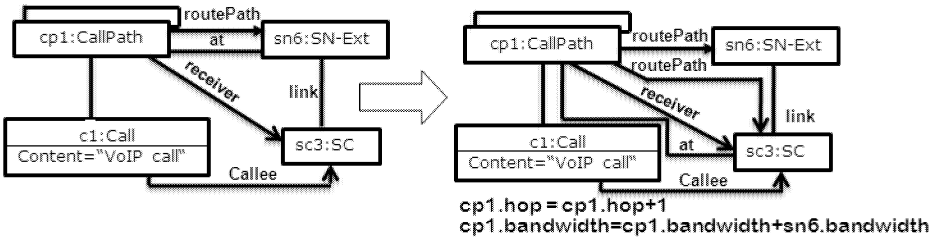


Figure 5: Forwarding packets to callee skype client

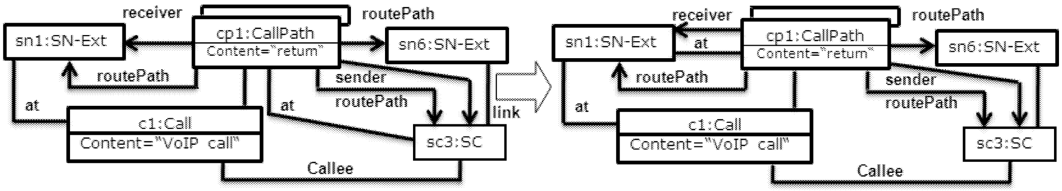


Figure 6: Return all packets to host super node

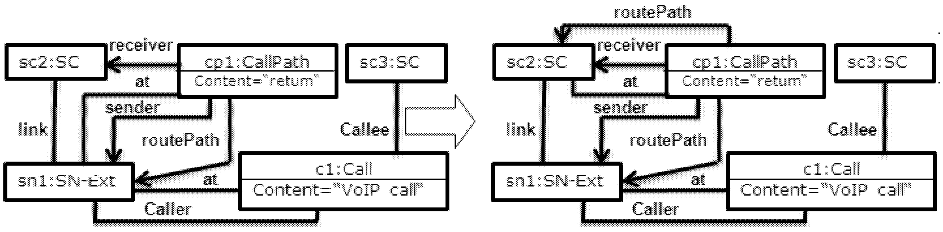


Figure 7: Forward detail of best path to the caller Skype client

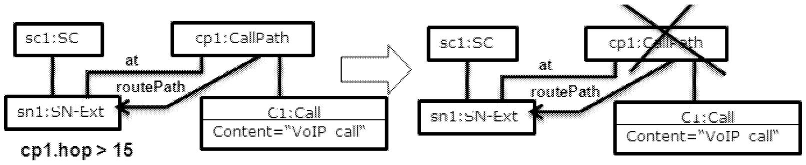


**Figure 8: Delete packets**

**Table 1: Computation of best overlay routing path**

| Bandwidth<br>(56k to 20000k) | Latency<br>(150ms to 300ms) | HOPs<br>(1 to 15) | $B_N$ | $L_N$ | $H_N$ | Score |
|---|---|---|---|---|---|---|
| 9235 | 199 | 1 | 0.460 | 0.327 | 0.000 | 2.522 |
| 11254 | 247 | 3 | 0.561 | 0.647 | 0.143 | 2.430 |
| 16439 | 261 | 2 | 0.821 | 0.740 | 0.071 | 2.894 |
| **19721** | **233** | **3** | **0.986** | **0.553** | **0.143** | **3.525** |
| 122 | 190 | 6 | 0.003 | 0.267 | 0.357 | 1.862 |
| 9564 | 286 | 11 | 0.477 | 0.907 | 0.714 | 1.448 |
| 11324 | 171 | 4 | 0.565 | 0.140 | 0.214 | 2.737 |

**Rule in Figure 7: Forward details of best path to the caller SC**. This rule forwards the actual path details for transmission of voice traffic to twords callee client. The path is the best path among the available overlay path computed based on the objective function. This path will be used by caller to transmit packets to the destination callee.

**Rule in Figure 8: Delete packets**. This rule deletes all those packets whose *HOP* value is more than *15*. Thus, it prevents the circulation of packets in the network for indefinite periods. The deletion of packets is shown by the double crossed graph node.

## 5. CONCLUSION AND FUTURE WORK

This paper illustrates the ongoing work on modeling VoIP networks using graph transformation. We are using bandwidth, latency and hop count to select the best routing path in the overlay network based on the objective function. Future work will address the simulation of the model using stochastic simulator and NS-2. The stochastic simulations of the graph-transformation system will be compared with NS-2 based simulations, so that the comparison of different simulation approaches can yield new insights into the suitability of different simulation paradigms for various types of analysis of such systems.

## 6. REFERENCES

[1] R. Gupta, A. K. Somani. Pricing Strategy for Incentivizing Selfish Nodes to Share Resources in Peer-to-Peer (P2P) Networks . Proc.of12th IEEE Int. Conf.on Networks (ICONŠ04) 2:624Ű629, 2004.

[2] M.J. Arif, S. Karunasekera, S. Kulkarni. SOVoIP: True Convergence of Data and Voice Network. 16th ACM WWW confrence Banff, Alberta, Canada, 2007.

[3] R. Heckel. Stochastic Analysis of Graph Transformation Systems: A Case Study in P2P Networks. In Proc. Intl. Colloquium on Theoretical Aspects of Computing(ICTACŠ05). LNCS 3722, pp. 53Ű69. Springer-Verlag, 2005.

[4] O. Lysne, J. M. Montanana, T. M. Pinkston. Simple Deadlock-Free Dynamic Network Reconfiguration. LNCS, SpringerLink 3296/2005:504Ű515, 2004.

[5] Shansi Ren, Lei Guo, Xiaodong Zhang.ASAP: An AS-Aware Peer-relay protocol for high quality VoIP. Proc. of 26th Int. Conf. on Distributed Computing Systems (ICDCS'06), Lisbon, Portugal, July 4-7, 2006

[6] Network Simulator-NS-2

[7] Juan de Lara et al. Attributed Graph Transformation with Node Type Inheritance. Theor. Comput. Sci. In Fundamental Aspects of Software Engineering, Vol. 376, No. 3. (15 May 2007), pp. 139-163.

[8] S. Guha, N. Daswani, R. Jain. An Experimental Study of the Skype Peer-to-Peer VoIP System. In IPTPSŠ06: The 5th International Workshop on Peer-to-Peer Systems, 2006.

[9] A. Khan, P. Torrini, R. Heckel. Model-based Simulation of VoIP Netowrk Reconfiguration using Graph Transformation System, Vol.17 EASST, ICGT, 2008.

[10] International Telecommunication Union, http://www.itu.int

[11] G. Beliakov, A. Pradera, T. Calvo. Aggregation Functions: A Guide for Practitioners. Springer, 2007.

[12] L. Ji. Computation in Peer-to-Peer Networks. Department of computer Scince, Univ. of Saskatchewan, Canada.

[13] A. Markopoulou, F. Tobagi, and M. Karam. Assessing the Quality of Voice Communications Over Internet Backbones. IEEE/ACM Trans. Netw., 11(5), 2003.

[14] T. Nguyen and A. Zakhor. Path Diversity with Forward Error Correction (pdf) system for Packet Switched Networks. In Proceedings of IEEE INFOCOMŠ03.

[15] S. Tao et al. Improving VoIP Quality through Path Switching. In Proceedings of IEEE INFOCOMŠ05.

[16] Zhu, Y., Dovrolis, C., and Ammar, M.Dynamic Overlay Routing Based on Available Bandwidth Estimation: A Simulation study. Comput. Netw. 50, 6 (Apr. 2006), 742-762.

[17] D. Rossi, M. Mellia, M. Meo. Following Skype Signaling Footsteps. Telecommunication Networking Workshop on QoS in Multiservice IP Networks, 2008.

# Microarray data and image simulation

Ignazio Infantino
Carmelo Lodato
Salvatore Lopes
ICAR - Istituto di Calcolo e Reti ad Alte Prestazioni, branch of Palermo
CNR - Consiglio Nazionale delle Ricerche
Viale delle Scienze, edificio 11,
90128, Palermo
Italy
E-mail: infantino@pa.icar.cnr.it - c.lodato@pa.icar.cnr.it - s.lopes@pa.icar.cnr.it

**KEYWORDS**
Microarray images, spot segmentation, image simulation.

## ABSTRACT

Microarray technologies have become very popular among biologists because a single experiment can produce a large amount of gene expression data. However, the experimental process, from slide manufacturing to data analysis, can be affected by random errors and/or systemic bias caused by several factors. Moreover, an evaluation of the data analysis tools can not be objectively performed because of the lack of specific benchmarks. The paper describes a method of data and microarray image simulation that provides researchers with reference tests for evaluating the performance of analysis tools. The relevant characteristics of a specific experiment are used for generating synthetic microarray images and the corresponding gene expression values. Hence, one can estimate expected errors comparing simulated data to the results of different analysis packages and then can choose the most suitable technique for the specific experiment. Three different segmentation algorithms have been tested on simulated images. The results on the classification correctness of gene activation are reported.

## INTRODUCTION

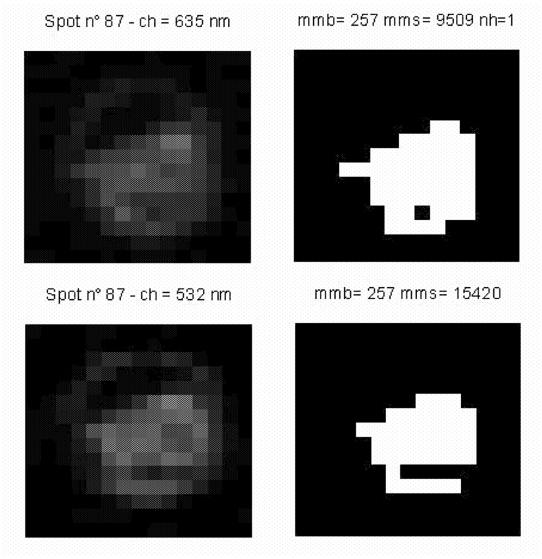
DNA Microarray is a technology for measuring the amount of mRNA produced by genes thus evaluating their activity level in biological samples. This technology has large diffusion in biological research field (Xiang and Chen 2000), but the evaluation of employed computational approaches is still an open problem (Kothapalli et al. 2002). A model of microarray experiment could be used for producing synthetic data as reference test but one has to face a lot of issues (Quackenbush 2001). For example, size and shape of spots, presence of scratches and defects on sub layer (Fraley et al. 2005), spot grid misalignment, noise (Cho and Lee 2004; Rocke and Durbin 2001; Balagurunathan 2004 et al.) and so on are highly variable and they also depend on the current technology. Even image segmentation algorithms affect results (Ahmed 2004 et al.) or produce significant differences on them (Yang 2002 et al.). Taking into account all error sources is almost infeasible, and a too complex model would be hardly adaptable and manageable to simulate specific experiment conditions. Moreover, technical improvements change the appearance of the produced microarray images and then we need to find approaches that can be flexible and adaptable both in respect to old and new technologies. A promising approach is to create controlled simulated images (Nykter et al. 2006; Balagurunathan et al. 2002), in order to perform the evaluation of results obtained by a given analysis software. Several studies on DNA microarray experiment simulation can be found in literature: noise analysis (Tu et al. 2002), pixel intensity statistical distribution (Davies and Seale 2005), critical parameters of image analysis (Wierling et al. 2002) and expression level statistical analysis (Singhal et al. 2003). Here, we present a method for the simulation of microarray data using the characteristics extracted from real experiments. The novel aspect of our work consists in generating both image and gene expression data thus creating a reference test that can be used to analyze and to effectively evaluate the performance of microarray image analysis tools. Hence, we can choose the most suitable technique for the real experiment.

## ANALYSIS OF REAL MICROARRAY IMAGES

There are several factors affecting cDNA experiments, i.e. mRNA preparation, PCR amplification, transcription, labeling, array geometry, target volume, target linking, hybridization, sub-layer non-homogeneity, microarray scanning and image analysis. We are interested in evaluating the influence on the results of analysis tools applied to the scanned images. Hence, we need to derive general and local image properties in order to set up adequate and effective parameters for a spot simulation model. We can use data stored in public databases for deriving the parameter values necessary to the spot and grid generation. In the present work, we choose the Stanford MicroArray Database as data source. For each experiment in SMD, it is possible to download raw data and the results reported by experimenters. The images, scanned at 635 nm and 532 nm wavelengths, are the starting point for the real microarray experiment analysis (Yang et al. 2001). Grid geometry, spot locations, and other low level details are recovered from the results file. A square region around each feature is considered for analyzing background and spot properties. Independently from the results saved by experimenters and the used normalization method, we calculate the following values for each feature region:

- median, mean, and variance of background pixel intensity of each channel;
- median, mean, and variance of spot pixel intensity of each channel;
- spot geometric center coordinates;
- number of holes and their location;
- correlation of spot pixels of two channels.

Pixels of the considered area are classified in two clusters, spot and background, by a K-means algorithm that uses the squared Euclidean distance as metric (Kanungo et al. 2002). In this way, we implicitly find a threshold value that depends on the local brightness of the considered spot area. K-means initial seed points are the upper-left and the central pixel of the region. Background classified regions but included inside the spot area are considered as holes. The first column of Figure 1 shows the considered spot area in the two channels where brightness intensity is rescaled for a better visualization. The second column shows segmented area by K-means algorithm and reports some values: mmb is median of background pixels, mms is median of spot pixels, nh is the number of holes within spot.



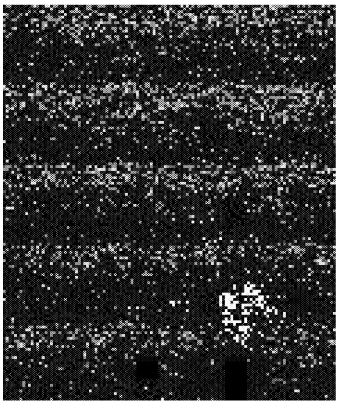
Figures 1: Experiment #58928 Analysis of Spot n° 87

We also derive a map where a representative color is assigned to each spot for capturing spatial distribution patterns of gene activation. Each pixel on the map corresponds to a spot of the real microarray and its color gives a rough representation of the gene activation: black correspond to little or not activated genes, red and green respectively to down-regulated and over-regulated genes, yellow to almost equally expressed genes (see example in Figure 2). Regarding noise arising from various sources, the calculated relevant characteristics are:

- median of background pixel intensities medians, MMb_ch1, MMb_ch2;
- mean of background pixel intensities variances, Varb_ch1, Varb_ch2.

Moreover, the spot alignment is not perfectly regular and then we estimate the following quantities:

- mean of x misalignment between geometric center of spot and centre of sub-region which includes it Mx_mis_ch1, Mx_mis_ch2;
- mean and variance of y misalignment, My_mis_ch1, My_mis_ch2.



Figures 2: Spatial Distribution Example of Gene Activation

Finally, for what concerns holes within spots, the following quantities are considered:

- mean of number of holes Mn_hole_ch1, Mn_hole_ch2;
- mean of distances of holes from spot geometric center Md_hole_ch1, and Md_hole_ch2.

We also developed a procedure to highlight eventual large defects. They are characterized by a partial or total removal of the sensitive sub-layer and correspond to dark areas in scanned images and can be detected as anomalous grouping of missing spot. An example is depicted in Figure 2 where a large defect is highlighted by white pixels.

## SIMULATION MODEL

From a visual analysis of microarray images it can be realized that spots generally do not have a regular shape. In fact, they often present some morphological distortion instead of being circular (Tu et al. 2002). For example, spots with doughnut shape are frequently observed in real microarray images. Such a shape can be effectively modeled by a linear combination of two bivariate Gaussian distributions (Braendle et al. 2003; Ekstroem et al 2004). The developed model, relying just on the above consideration, allows to get a wide set of spot shapes. In fact, a generic shape can be obtained as envelope of several surfaces. It has been assumed that a spot is entangled in a square region of fixed size. Such a region is the definition domain *D* of all the surfaces the envelope is made of. Given a bivariate Gaussian distribution *G(x,y)* and a set of *n* secant planes $\{P_n(x,y)\}$, each surface can be expressed by the last element of a sequence $\{C_n(x,y)\}$ of linear combination between functions describing a surface and the corresponding secant plane. The first element of the sequence $C_0(x,y)$ coincides with the surface described by the given Gaussian distribution $G(x,y)$ while the generic element is a linear combination between the previous one and the corresponding secant plane. Such a procedure has to be repeated for each envelope surface. Hence, denoting with $m$ the generic surface, the model can be expressed by the following equations:

$$C_{m0}(x,y) = G_m(x,y)$$

$$C_{mn}(x,y) = C_{mn-1}(x,y) \qquad (2)$$

$$\forall x, y \in D \mid C_{mn-1}(x,y) \leq P_{mn}(x,y)$$

$$C_{mn}(x,y) = a_{mn}C_{mn-1}(x,y) + b_{mn}P_{mn}(x,y) \qquad (3)$$

$$\forall x, y \in D \mid C_{mn-1}(x,y) > P_{mn}(x,y)$$

with $a_{mn}, b_{mn} \in \{-1,0,1,2\}$ where

$m$=1,2,...$M$. $M$ no. envelope surfaces

$n$=1,2,...$N_m$. $N_m$ no. section planes of $m$-th surface

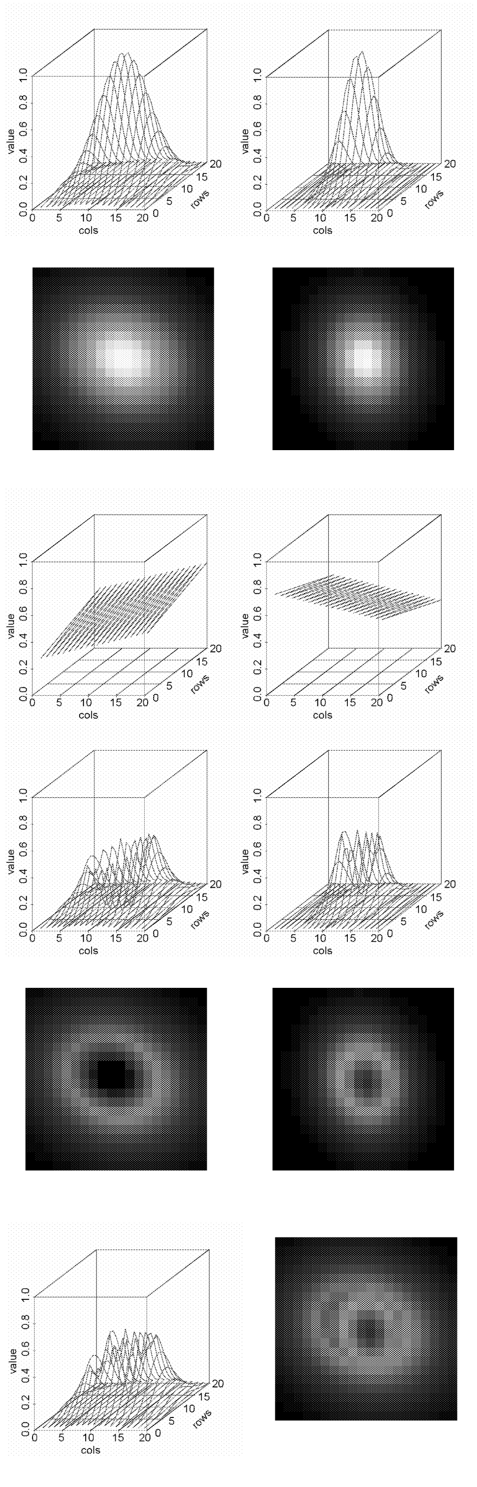
The values of coefficients $a_{mn}$ and $b_{mn}$, affect the specific type of shape, convex, plane or concave, that will be obtained. Equations (2) and (3) describe how the surface is modified by the application of the corresponding secant plane. In more detail, the surface is generally cut in two regions. The one under the secant plane remain unchanged, whereas the other is modified in accord to the equation (3). There are three different combination allowed modes, namely *clip* ($a_{mn}$ = 0, $b_{mn}$ = 1), *dig* ($a_{mn}$ = -1, $b_{mn}$ = 2) and *lift* ($a_{mn}$ = 2, $b_{mn}$ = -1). The above described procedure is repeated until all the secant planes have been applied, thus producing a single element of the envelope. The succeeding elements will be defined repeating the same procedure as many times as many are the number of specified Gaussian. Hence, the final surface is obtained enveloping all the single surfaces:

$$S(x,y) = \max\left(C_{mN_m}(x,y)\right) \qquad (4)$$

The resulting function $S(x,y)$ represents the spatial distribution of the simulated spot brightness. Obviously, the generated spot shape will depend on the number of Gaussian surfaces and the corresponding set of secant planes. The parameters characterizing each Gaussian distribution are the standard deviation along $x$-, $y$-direction ($\sigma_x$,$\sigma_y$), the correlation factor $\rho$, and the peak coordinates ($peak_x$, $peak_y$), whereas at least three coefficients ($a$, $b$, $c$) are needed in order to specify each correlated secant plane. Some of the input parameters are derived from the real experiment. Among them, there is the peak location of each Gaussian distribution, as well as, the standard deviations and the correlation factor. All these values affect location, radius and orientation of the generated spot and could make its shape more or less elliptic. A further parameter specifies the desired peak value of spot brightness distribution. Just as for an example, we report the spot resulted by the envelope of two bivariate Gaussian, each modified in dig mode by a single secant plane. The parameters used in this example are summarized in Table 1 where the Gaussian peak coordinates and the coefficients of the secant plane equations are referred to the bottom left corner of the spot area.

Table 1: Parameter Values for the Spot Generation

| | |
|---|---|
| Spot area | 20 x 20 pixel |
| Gaussian no. 1 | $s_x$=3.5, $s_y$=4.0, ρ=0.15<br>*peak*$_x$*=10.0, peak*$_y$*=10.0* |
| Secant plane | $a$=0.01, $b$=0.01, $c$=0.45 |
| Dig mode parameters | $a_{11}$=-1, $b_{11}$=2 |
| Gaussian no. 2 | $s_x$=3.0, $s_y$=2.5, ρ=0.10<br>*peak*$_x$*=9.9, peak*$_y$*=9.8* |
| Secant plane | $a$=-0.01, $b$=-0.01, $c$=0.55 |
| dig mode parameters | $a_{11}$=-1, $b_{11}$=2 |



Figures 3: Spot Generation Phases

The complete operative sequence of spot production is presented in Figure 3 where the surfaces obtained by cutting the Gaussians with the respective secant planes, the corresponding spots and the final envelope are shown. The procedure for microarray image generation consists in generating spots one at time and assembling them in accord to the microarray layout specified by the input parameters. The layout parameters are typically the total spot and macro block number, the spot and block per row and column number, the distance between adjacent spots and blocks. Each spot is generated in accord to a set of input parameters derived from the corresponding features of real experiment. The following step consists in giving a virtual color to the spot in order to generate both red and green channel according to the map described in previous section. Hence, the simulated brightness is distributed between the red and green channel still preserving the appearance of the real spot. The final image will then be produced assembling all the generated spots. Noise can be applied to the image in two different ways: a small variation of brightness at most image pixels (distributed noise); a significant brightness variation applied to a few pixels only (local noise). A report, including all the parameter values together with the characteristics of generated spots necessary to reproduce the image, is also compiled. Moreover, a binary map, where the pixels belonging to the spots are identified against the background, will be stored for each spot.

## RESULTS AND DISCUSSION

We have performed various experiments using data extracted from Stanford Microarray Database for testing three different spot extraction algorithms implemented in commercial analysis packages. In the following we report the results concerning the experiment labeled as #52928 in SMD. It is important to notice that we report these results for showing a possible use of the proposed method and not for evaluating the goodness of our approach that derives from the uncertainty and variability characterizing real experiments instead. The experiment slide contains 41472 spots and the scanned image size is 5552 x 1912 pixels. Spot properties are extracted using the methodology previously explained. They are summarized by the following quantities:

- median of spot pixel intensities medians, $MM_{spot\_ch1}$, $MM_{spot\_ch2}$;
- mean of spot pixel intensities variances, $Var_{spot\_ch1}$, $Var_{spot\_ch2}$;
- mean of correlation of spot pixel intensities between channels, $Corr_{spot\_ch1}$, $Corr_{spot\_ch2}$;
- variance of correlation of spot pixel intensities between channels, $VarCorr_{spot\_ch1}$, $VarCorr_{spot\_ch2}$.

Calculated values of the relevant features for the experiment #52928 are reported in Table 2. The list of main parameters necessary to the simulation of a generic spot is reported in Table 3. Most values are calculated from the features of the corresponding real spot while the others are estimated in accord to the statistical properties of the real microarray. Distributed and local noise has been applied to the simulated image. Distributed noise occurs with a probability of 99.95 %, whereas local noise occurs with a probability of 0.05 %. The distributed noise modifies the pixel brightness by a random value extracted from a range -0.1~0.2. The random variation caused by the local noise *Nlv* lies in the range 0.5~2.0.

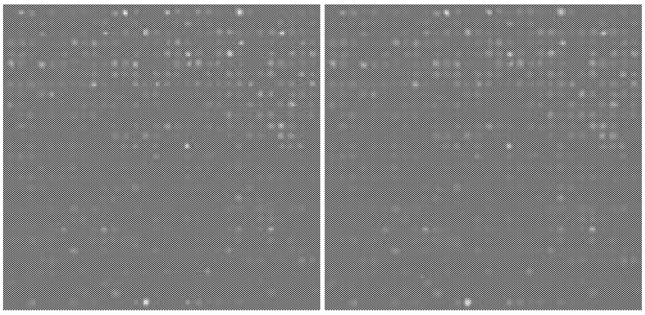
Table 2: Calculated Values for Experiment #52928

| Name | Ch 635 nm | Ch 532 nm |
|---|---|---|
| $MM_{spot\_ch}$ | 2268 | 3890 |
| $Var_{spot\_ch}$ | 3321.1 | 4272.1 |
| $Corr_{spot\_ch1\_ch2}$ | 0.8363 | |
| $VarCorr_{spot\_ch1\_ch2}$ | 0.0199 | |
| $MM_{b\_ch}$ | 255 | 953 |
| $Var_{b\_ch}$ | 447.4 | 1113.6 |
| $M_{x\_mis\_ch}$ | 0.025 | 0.023 |
| $M_{y\_mis\_ch}$ | 0.032 | 0.038 |
| $M_{n\_hole\_ch}$ | 0.67 | 0.76 |
| $M_{d\_hole\_ch}$ | 1.32 | 1.32 |

Table 3: Microarray Generation Parameters List

| Name | Description |
|---|---|
| *SaY, SaX* | Height and width of the spot area |
| *Nga, Np* | No. of gaussian and secant planes |
| *Gpk* | Peak value of the gaussian distribution |
| *GpkY, GpkX* | Vertical and horizontal shift from the spot area centre |
| $s_x$, $s_y$, $\rho$ | Standard deviations and correlation factor |
| *a, b, c* | x, y coefficient and known term of plane equation |
| *Pdes* | DE spot percentage |
| *Ndp, Nlp* | Distributed and local noise probability |
| *Ndv, Nlv* | Distributed and local noise variation |
| *Nrf* | Noise repartition factor between channels |

The first simulated block and the corresponding real block of the experiment #52928 are shown in figure 4.



Figures 4: Simulated and Real First Block

Three different segmentation algorithms have been applyed to both channels of the synthetic image: HS - histogram based segmentation (Chen et al. 1997); ACS - adaptive circle segmentation (GenePix); OS - Otsu segmentation (Otsu 1979). In order to estimate the correctness of results obtained by each method, we adopted the following criterion. We classified the simulated spots in four classes: not processed (their brightness is very low and they are almost undetectable); over-regulated (log ratio > 1); down-regulated (log ratio < -1); equally expressed (-1 < log ratio < 1).

For each method we calculate the percentage of correctly classified spot. The results are summarized in Table 4. In particular, a matrix for evaluating the correctness of spot activation is reported for each approach. Given a row of such matrix we report how the spots of a given class are distributed by the algorithm in exam. For example, the second row of Histogram approach evaluation tell us that simulated spots having log ratio greater than 1.0 are corrected classified only in the 49.82% of the cases.

Table 4: Spot Classification for the Simulated Experiment

| **True vs processed** | **Not processed** | **LogR>1** | **LogR<-1** | **-1<LogR<1** |
|---|---|---|---|---|
| *Histogram error= 1.3794* | | | | |
| **Not processed** | 0.9962 | 0.0036* | 0.0001* | 0.0001 |
| **LogR>1** | 0.5018* | 0.4982 | 0.0000* | 0.0000* |
| **LogR<-1** | 0.5538* | 0,0007* | 0.4428 | 0.0027* |
| **-1<LogR<1** | 0.2977 | 0.3167* | 0.0000* | 0.3856 |
| *Circle error= 1.4290* | | | | |
| **Not processed** | 0.9999 | 0.0001 | 0.0000 | 0.0000 |
| **LogR>1** | 0.6012 | 0.3988 | 0.0000 | 0.0000 |
| **LogR<-1** | 0.6777 | 0.0027 | 0.3089 | 0.0108 |
| **-1<LogR<1** | 0.5847 | 0.1365 | 0.0000 | 0.2788 |
| *Otsu error= 0.9478* | | | | |
| **Not processed** | 0.9748 | 0.0207 | 0.0019 | 0.0001 |
| **LogR>1** | 0.4123 | 0.5866 | 0.0000 | 0.0011 |
| **LogR<-1** | 0.2194 | 0.000 | 0.7712 | 0.0094 |
| **-1<LogR<1** | 0.0032 | 0.283 | 0.0000 | 0.7130 |

The erroneous classifications are all in the class of *Not processed* spots 50.18%. Moreover the table reports an error index (values from range 0.0 ~ 4.0) as sum of relevant classification errors (values marked with stars). Naturally, the diagonal elements of the matrix express the correctness of the approach. In this example, real experiment data were evaluated by an ACS based tool, but it is clear that an OS based tool could have assured a better quality of results. If we have to conduct a new experiment with similar or identical condition, we can evaluate which is the more precise approach, or we can have an idea of error magnitude conditioning our results.

## CONCLUSIONS

Our research is focused on the enhancement of the reliability of microarray experiment results. In particular, we are interested in the image analysis phase because there is a general lack of reference tests for calibrating or evaluating the segmentation algorithms used by microarray software packages. Our idea is that does not exist a segmentation algorithm or software definitely better than all the others, but it is possible to choose the most suitable for given experiment. The proposed method allows to create simulated images of a given real microarray experiment. These images and corresponding data can be used as reference test for the analysis tools employed in this technology. The implemented spot simulation model uses parameters that are derived from real experimental data. As further work we are also interested in a suitable and reliable technique for evaluating the similarity between the real and simulated microarray images. The proposed method has been implemented in R and a web version is currently under development.

## REFERENCES

Ahmed A. A.; M. Vias; N. G. Iyer; C. Caldasand J. D. Brenton. 2004. "Microarray segmentation methods significantly influence data precision". *Nucleic Acids Research*, 32(5):e50.

Braendle N.; H. Bishof and H. Lapp. 2203. "A generic and robust DNA microarray image analysis". *Machine Vision and Applications*, 15:11-28.

Balagurunathan Y.; E. R. Dougherty; Y. Chen; M. L. Bittner and J. M. Trent. 2002. "Simulation of cDNA microarrays via a parameterized random signal model". *Journal Biomed Opt.*, 7(3):507-523.

Balagurunathan Y.; N. Wang; E. R. Dougherty; D. Nguyen; Y. Chen; M. L. Bittner; J. M. Trent and R. Carroll. 2004. "Noise factor analysis for cDNA microarrays". *Journal Biomed Opt*, 9(4):663-678.

Chen Y.; E. R. Dougherty and M. L. Bittner. 1997. "Ratio-based decisions and the quantitative analysis of cDNA microarray images". *Journal of Biomedical Optics*, 2:364-374.

Cho H. and J. K. Lee. 2004. "Bayesian hierarchical error model for analysis of gene expression data". *Bioinformatics*, 20(13):2016-2025.

Davies S. W. and A. Seale. 2005. "DNA Microarray Stochastic Model". *IEEE Transaction on Nanobioscience*, 4(3):248-254.

Ekstroem C. T.; S. Bak; C. Kristensen and M. Rudemo. 2004. "Spot shape modeling and data transformations for microarrays". *Bioinformatics*, 20(14):2270-2278.

Kanungo T.; D. M. Moun; N. Netanyahu; C. Piatko; R. Silverman and A. Y. Wu. 2002. "An efficient k-means clustering algorithm: Analysis and implementation". *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24:881-892.

Kothapalli R.; S. J. Yoder; S. Mane and T. P. Jr Loughran. 2002. "Microarray results: how accurate are they?". *BMC Bioinformatics*, 3:22.

Li Q.; C. Fraley; R. E. Bumgarner; K. Y. Yeung and A. E. Raftery. 2005. "Donuts, scratches and blanks: robust model-based segmentation of microarray images". *Bioinformatics*, 21(12):2875-2882.

Nykter M.; T. Aho; M. Ahdesmaki; P. Ruusuvuori; A. Lehmussola and O. Yli-Harja. 2006. "Simulation of microarray data with realistic characteristics". *BMC Bioinformatics*, 7:349.

Otsu N. 1979. "A threshold selection method from gray-level histogram". *IEEE Transaction on System Man and Cybernetics*, 8:62-66.

Quackenbush J.. 2001. "Computational analysis of microarray data". *Nat. Rev. Genet.*, 2, 418-427.

Rocke D. M. and B. Durbin. 2001. "A model for measurement error for gene expression array". *Journal Comput. Biol.*, 8(6):557-569.

Singhal S.; C. G. Kyvernitis; S. W. Johnson; L. R. Kaisera; M. N. Liebman and S. M. Albelda. 2003. "Microarray data simulator for improved selection of differentially expressed genes". *Cancer biology and therapy*, 2(4):383-391.

Tu Y.; G. Stolovitzky and U. Klein. 2002. "Quantitative noise analysis for gene expression microarray experiments". in *Proceedings of National Academy of Sciences of USA*, edited by Austin R. H., Princeton: Princeton University, 99(22):14031-14036.

Xiang C. C. and Y. Chen. 2000. "cDNA Microarray Technology and its Applications". *Biotechnology Advances*, 18, 35-46.

Yang Y.H.; M. J. Buckley and T. P. Speed. 2001. "Analysis of cDNA microarray images". *Briefings in bioinformatics*, 2(4):341-349.

Yang Y. H.; M. J. Buckley; S. Dudoit and T. P. Speed. 2002. "Comparison of methods for image analysis on cDNA microarray data". *Journal of Computational and Graphical Statistics*, 11:108-136.

Wierling C. K.; M. Steinfath; T. Elge; S. Schulze-Kremer; P. Aanstad; M. Clark; H. Lehrach and T. Herwig. 2002. "Simulation of DNA array hybridization experiments and evaluation of critical parameters during subsequent image and data analysis". *BMC Bioinformatics*, 3:22.

# DATA PLACEMENT AND MIGRATION STRATEGIES FOR VIRTUALISED DATA STORAGE SYSTEMS

H.A. Bond N.J. Dingle F. Franciosi P.G. Harrison W.J. Knottenbelt
Department of Computing, Imperial College London, South Kensington Campus, London SW7 2AZ
Email: {hab06,njd200,ozzy,pgh,wjk}@doc.ic.ac.uk

**KEYWORDS**
Storage virtualisation, Data placement and migration

## ABSTRACT

This paper details a simulation intended to explore the viability of storage systems where data is placed intelligently depending on data-specific quality of service requirements and the characteristics of the underlying storage devices. Our implementation provides a framework for the automatic profiling of a virtualised storage system, specification of data-level quality of service requirements, simulation of data streams acting on the storage system and visualisation of the results. We demonstrate that an intelligent data placement algorithm can achieve a good match between desired and offered storage system quality of service.

## INTRODUCTION

The volume of data produced world-wide continues to grow. In the second quarter of 2008, shipped disk storage capacity was 1 777 petabytes, representing growth of 43.7% year on year[1]. Indeed, the International Data Corporation (IDC) forecasts that shipped disk storage capacity will increase at a compound annual rate of 38% for the next three years (Rydning and Reinsel, 2009).

Recently there has been a trend towards centralised storage systems, with the aim of reducing management overhead and lowering total cost of ownership. These systems are usually implemented in the form of a Virtualised Storage System (VSS), i.e. a contiguous logical volume where the underlying storage devices (e.g. solid state disks, RAID arrays, etc.) are abstracted away. Different devices provide very different storage characteristics, however, and their abstraction makes the job of efficiently placing data far more complex.

Performance is key to many users, but it is not the only characteristic to take into account when using VSSs. Reliability is another very important aspect, as is space efficiency. Throughout this paper we refer to these data considerations as Quality of Service (QoS). For example, a bank's transaction records have to be stored in a fashion that guarantees high reliability. A temporary file, by contrast, probably needs only high performance. Data placement is a very important aspect of this problem, but the QoS attributes and the patterns in which data is accessed are not static and might change over time for different data streams. This implies that not only initial allocation must be considered but also migration, to ensure delivered QoS remains as close to that required by the user as possible on a sustained basis.

A number of other projects have investigated similar ideas in terms of intelligent data placement. BORG (Block-reORGanization and Self-optimization in Storage Systems) (Bhadkamkar et al., 2009) is a module for the Linux kernel which sits between the file system layer (e.g. ext3, JFS, NTFS etc) and the I/O scheduler. It constantly evaluates process access patterns based on temporal, process-level, and block-level attributes and constructs access pattern graphs.

In Franciosi and Knottenbelt (2009), the authors present a modification of the well-known ext3 file system to include QoS attributes, which exploits redundant bits in the individual inodes to mark QoS attributes for sets of data. The system will use these attributes along with a performance profile of the underlying storage devices to intelligently place data.

This paper describes a simulation of a system which allocates and migrates data depending on a set of user-specified QoS requirements and the characteristics of the underlying VSS. We begin with a brief overview of modern storage systems, before detailing the design and implementation of our simulator. We describe how our simulation automatically profiles a VSS's logical address space to create a multi-attribute QoS model of the device, and then outline how the user specifies their QoS requirements and the characteristics of the data streams that will be used to drive the simulation. We further present the allocation and migration techniques that we have implemented, and show how the simulator visualises the resulting data layout and performance for the user. Finally, we present a number of case studies of the use of the simulator before concluding.

## BACKGROUND

Today the most common storage medium is the hard disk drive. This device consists of a mechanical read/write head which moves over a spinning disk made of fer-

[1] Source: IDC press release, 5 September 2008 (`http://www.idc.com/getdoc.jsp?containerId=prUS21411908`)

romagnetic material. Multiple hard disks can be used to improve performance and reliability, typically using one of the Redundant Array of Inexpensive Disks (RAID) levels. In this work we focus on the most commonly used configurations: RAID 0 (striping without redundancy), RAID 1 (mirroring), RAID 5 (distributed parity) and RAID 10 (stripes of mirrors). We note that our work could easily be extended to model additional schemes such as RAID 6 (double distributed parity).

| RAID Level | Performance | | Data Reliability | Space Efficiency |
|---|---|---|---|---|
| | Read | Write | | |
| 0 | H | H | L | H |
| 1 | H | L | H | L |
| 5 | M | L/M | M | M |
| 10 | M/H | L/M | H | L |

Table 1: Summary of RAID characteristics.

Each RAID level offers different trade-offs between performance, reliability and space efficiency, as summarised in Table 1. For example, RAID 0 offers very high performance as disk accesses can be performed in parallel and its space efficiency is also very good since it stores no redundant data. Its reliability is, however, poor as if one drive fails then data will be lost. On the other hand, RAID 1 offers very good reliability as it maintains a copy of all data so that if one drive is lost, all data on it can be recovered from the mirror. It also offers very good read performance as either disk can be accessed for a particular piece of data. Write performance is degraded, however, as a request must be serviced by both disks, and it is not particularly space-efficient as it requires twice the number of drives to store the same volume of data as RAID 0.

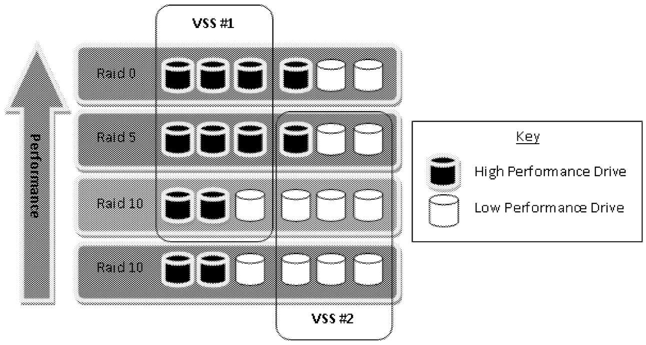


Figure 1: Two VSSs spread over four different storage tiers ordered by performance

It is not as easy to characterise VSSs in these terms, however. Figure 1 shows two VSSs spread over an infrastructure consisting of four different tiers, and that consequently provide different QoS attributes. VSS #1 will have a very good performance profile due to the majority of its underlying storage devices being of high performance and in fast RAID configurations. VSS #2, by contrast, does not provide such good performance but does provide good data reliability, as every single tier keeps redundancy data.

**Data Stream Characteristics**

| Data Stream A | Data Stream B |
|---|---|
| System critical<br>> 50% read requests<br>100% sequential<br>Requests typically small | Non-critical<br>100% write requests<br>100% non-sequential<br>Requests typically large |

Table 2: Example QoS requirements for 2 data streams.

The central concept of this work is that if we know the characteristics of the requests made to a VSS on a per-application basis (which we term a *data stream*) and also the user's QoS requirements for that stream, we can exploit intelligent data placement strategies to deliver better QoS. Consider the situation where Stream A and Stream B both want to access data on the same VSS. The QoS requirements for the two streams are given in Table 2. The most suitable place on the logical partition for data from Stream A is very likely to be different from the most suitable place for the data from Stream B.
We consider three QoS requirements:

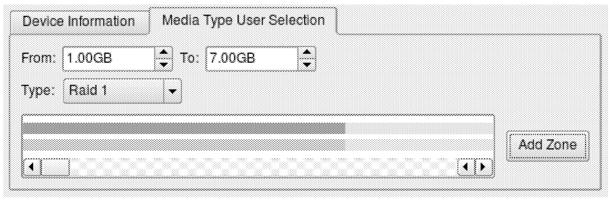
- Performance
- Reliability
- Space efficiency.

It is important to note that the QoS attributes for data are not static. We therefore consider not just the initial placement of data according to its requirements and the attributes of the underlying storage system, but also its migration within the storage system in response to changing access patterns and QoS requirements. Ideally this migration should be done automatically to minimise the management overhead of the VSS.

## PROFILING

In order to simulate a VSS faithfully, it is necessary to quantify the performance, reliability and space efficiency of every one of its (logical) block groups.
We characterise performance by throughput, which we measure in KB/s. We have implemented an automatic profiler which runs on a real VSS and issues block group-level requests and records the time that they take to complete. As input, we consider sequential and random accesses of two different request sizes (large and small) for both reads and writes, for a total of 8 possible combinations. The resulting times are then processed to yield the average throughput for each block group under each of these I/O patterns.

It is then necessary to classify the performance of areas of the underlying storage as either Low, Medium or High. These classifications should be relative to the underlying storage and not absolute since the profiler should not be dependent on the system it runs. We first find the throughput which lies on the $33^{rd}$ percentile and the $66^{th}$ percentile. Any throughput which lies below the $33^{rd}$ percentile is classified as Low, between the $33^{rd}$ percentile and the $66^{th}$ percentile is classified as Medium and anything greater than the $66^{th}$ percentile is classified as High.



From: 1.00GB To: 7.00GB

Type: Raid 1

Figure 2: The GUI for selecting device types within logical address ranges.

When profiling reliability and space efficiency, the abstraction of devices within a VSS is a problem since the profiler cannot automatically gather the required information. Instead we must rely on the user's knowledge of the makeup of the VSS, and we thus provide a tool to enable the user to manually choose the types of devices comprising the VSS and to specify the logical address range to which they correspond. Figure 2 shows an example of a VSS where the range 1GB-7GB describes a RAID 1 device and 7GB onwards describes a RAID 5 device. The coloured strips represent space efficiency (upper) and reliability (lower), where red corresponds to Low, yellow to Medium and green to High. From this and the RAID characteristics in Table 1, we automatically create a per block group classification of reliability and space efficiency in terms of High, Medium and Low.

## SIMULATION

In real systems a 'sea' of requests will be made to the storage from a variety of sources. For example, the system may be a file server containing users' home directories, while at the same time it may also house a database server storing experimental data. In order to simulate such situations we introduce the concept of data streams, and for each of these in a simulation the user must identify the following characteristics:

- Arrival Rate: the average number of I/O operations per unit time (assumed to arrive according to a Poisson process).
- Performance Required: High, Medium or Low
- Reliability Required: High, Medium or Low
- Space Efficiency Required: High, Medium or Low
- Data Size: The expected proportion of Small:Large data accesses
- Order of Access: The expected proportion of Sequential:Random data accesses
- I/O Type: The expected proportion of Read:Write data accesses

### Events and Event Generation

In order to achieve realistic simulation results there are three main points to consider:

- Which stream triggers the event
- What type of event (read or write) is triggered
- Amount of time until the next event

The likelihood of a particular stream being chosen depends on its arrival rate, and is calculated as $\frac{rate(stream)}{\sum_i rate(i)}$. After the stream has been selected, the type of I/O operation issued by that stream needs to be generated. These can be either reads or writes; we further distinguish between three types of write event: data creation, data deletion and data modification.

To drive the utilised capacity of the file system towards a target capacity, $t$, given space usage $c$, we require:

$$\begin{aligned} c > t &\longrightarrow P(delete) > P(create) \\ c = t &\longrightarrow P(delete) = P(create) \\ c < t &\longrightarrow P(delete) < P(create) \end{aligned}$$

We therefore use the following probabilities to calculate if a write operation is a modify, a create or a delete:

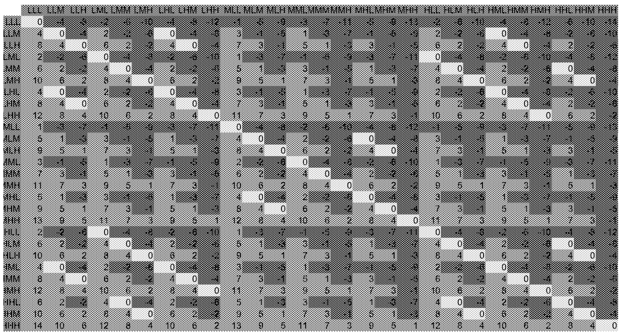
$$\begin{aligned} P(modify) &= k_{modify} \\ P(create) &= 1 - P(delete) \\ P(delete) &= \begin{cases} \frac{xk_{delete}}{2t} & c \leq t \\ \frac{(x-t)k_{delete}}{2(1-t)} + \frac{k_{delete}}{2} & otherwise \end{cases} \end{aligned}$$

The constants $k_{modify}$, $k_{create}$ and $k_{delete}$ are set to $\frac{2}{3}$, $\frac{1}{6}$ and $\frac{1}{6}$ respectively in the implementation.

It is assumed that events occur with exponentially-distributed inter-arrival times, with rate $\lambda = \sum_i rate(i)$.

### Intelligent Data Placement and Migration

The result of the profiling step is that we have a characterisation of every block group on the VSS in terms of performance, reliability and space efficiency. We also have matching QoS requirements for each data stream under the same three headings. The key challenge now is to place the data from the input streams on the VSS in such a way that these requirements are met as far as is possible. To achieve this we define and use a *Request-Receive* matrix.



| | LLL | LLM | LLH | LML | LMM | LMH | LHL | LHM | LHH | MLL | MLM | MLH | MML | MMM | MMH | MHL | MHM | MHH | HLL | HLM | HLH | HML | HMM | HMH | HHL | HHM | HHH |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LLL | 0 | -4 | -8 | -2 | -6 | -10 | -4 | -8 | -12 | -1 | -5 | -9 | -3 | -7 | -11 | -5 | -9 | -13 | -2 | -6 | -10 | -4 | -8 | -12 | -6 | -10 | -14 |
| LLM | 4 | 0 | -4 | 2 | -2 | -6 | 0 | -4 | -8 | 3 | -1 | -5 | 1 | -3 | -7 | -1 | -5 | -9 | 2 | -2 | -6 | 0 | -4 | -8 | -2 | -6 | -10 |
| LLH | 8 | 4 | 0 | 6 | 2 | -2 | 4 | 0 | -4 | 7 | 3 | -1 | 5 | 1 | -3 | 3 | -1 | -5 | 6 | 2 | -2 | 4 | 0 | -4 | 2 | -2 | -6 |
| LML | 2 | -2 | -6 | 0 | -4 | -8 | -2 | -6 | -10 | 1 | -3 | -7 | -1 | -5 | -9 | -3 | -7 | -11 | 0 | -4 | -8 | -2 | -6 | -10 | -4 | -8 | -12 |
| LMM | 6 | 2 | -2 | 4 | 0 | -4 | 2 | -2 | -6 | 5 | 1 | -3 | 3 | -1 | -5 | 1 | -3 | -7 | 4 | 0 | -4 | 2 | -2 | -6 | 0 | -4 | -8 |
| LMH | 10 | 6 | 2 | 8 | 4 | 0 | 6 | 2 | -2 | 9 | 5 | 1 | 7 | 3 | -1 | 5 | 1 | -3 | 8 | 4 | 0 | 6 | 2 | -2 | 4 | 0 | -4 |
| LHL | 4 | 0 | -4 | 2 | -2 | -6 | 0 | -4 | -8 | 3 | -1 | -5 | 1 | -3 | -7 | -1 | -5 | -9 | 2 | -2 | -6 | 0 | -4 | -8 | -2 | -6 | -10 |
| LHM | 8 | 4 | 0 | 6 | 2 | -2 | 4 | 0 | -4 | 7 | 3 | -1 | 5 | 1 | -3 | 3 | -1 | -5 | 6 | 2 | -2 | 4 | 0 | -4 | 2 | -2 | -6 |
| LHH | 12 | 8 | 4 | 10 | 6 | 2 | 8 | 4 | 0 | 11 | 7 | 3 | 9 | 5 | 1 | 7 | 3 | -1 | 10 | 6 | 2 | 8 | 4 | 0 | 6 | 2 | -2 |
| MLL | 1 | -3 | -7 | -1 | -5 | -9 | -3 | -7 | -11 | 0 | -4 | -8 | -2 | -6 | -10 | -4 | -8 | -12 | -1 | -5 | -9 | -3 | -7 | -11 | -5 | -9 | -13 |
| MLM | 5 | 1 | -3 | 3 | -1 | -5 | 1 | -3 | -7 | 4 | 0 | -4 | 2 | -2 | -6 | 0 | -4 | -8 | 3 | -1 | -5 | 1 | -3 | -7 | -1 | -5 | -9 |
| MLH | 9 | 5 | 1 | 7 | 3 | -1 | 5 | 1 | -3 | 8 | 4 | 0 | 6 | 2 | -2 | 4 | 0 | -4 | 7 | 3 | -1 | 5 | 1 | -3 | 3 | -1 | -5 |
| MML | 3 | -1 | -5 | 1 | -3 | -7 | -1 | -5 | -9 | 2 | -2 | -6 | 0 | -4 | -8 | -2 | -6 | -10 | 1 | -3 | -7 | -1 | -5 | -9 | -3 | -7 | -11 |
| MMM | 7 | 3 | -1 | 5 | 1 | -3 | 3 | -1 | -5 | 6 | 2 | -2 | 4 | 0 | -4 | 2 | -2 | -6 | 5 | 1 | -3 | 3 | -1 | -5 | 1 | -3 | -7 |
| MMH | 11 | 7 | 3 | 9 | 5 | 1 | 7 | 3 | -1 | 10 | 6 | 2 | 8 | 4 | 0 | 6 | 2 | -2 | 9 | 5 | 1 | 7 | 3 | -1 | 5 | 1 | -3 |
| MHL | 5 | 1 | -3 | 3 | -1 | -5 | 1 | -3 | -7 | 4 | 0 | -4 | 2 | -2 | -6 | 0 | -4 | -8 | 3 | -1 | -5 | 1 | -3 | -7 | -1 | -5 | -9 |
| MHM | 9 | 5 | 1 | 7 | 3 | -1 | 5 | 1 | -3 | 8 | 4 | 0 | 6 | 2 | -2 | 4 | 0 | -4 | 7 | 3 | -1 | 5 | 1 | -3 | 3 | -1 | -5 |
| MHH | 13 | 9 | 5 | 11 | 7 | 3 | 9 | 5 | 1 | 12 | 8 | 4 | 10 | 6 | 2 | 8 | 4 | 0 | 11 | 7 | 3 | 9 | 5 | 1 | 7 | 3 | -1 |
| HLL | 2 | -2 | -6 | 0 | -4 | -8 | -2 | -6 | -10 | 1 | -3 | -7 | -1 | -5 | -9 | -3 | -7 | -11 | 0 | -4 | -8 | -2 | -6 | -10 | -4 | -8 | -12 |
| HLM | 6 | 2 | -2 | 4 | 0 | -4 | 2 | -2 | -6 | 5 | 1 | -3 | 3 | -1 | -5 | 1 | -3 | -7 | 4 | 0 | -4 | 2 | -2 | -6 | 0 | -4 | -8 |
| HLH | 10 | 6 | 2 | 8 | 4 | 0 | 6 | 2 | -2 | 9 | 5 | 1 | 7 | 3 | -1 | 5 | 1 | -3 | 8 | 4 | 0 | 6 | 2 | -2 | 4 | 0 | -4 |
| HML | 4 | 0 | -4 | 2 | -2 | -6 | 0 | -4 | -8 | 3 | -1 | -5 | 1 | -3 | -7 | -1 | -5 | -9 | 2 | -2 | -6 | 0 | -4 | -8 | -2 | -6 | -10 |
| HMM | 8 | 4 | 0 | 6 | 2 | -2 | 4 | 0 | -4 | 7 | 3 | -1 | 5 | 1 | -3 | 3 | -1 | -5 | 6 | 2 | -2 | 4 | 0 | -4 | 2 | -2 | -6 |
| HMH | 12 | 8 | 4 | 10 | 6 | 2 | 8 | 4 | 0 | 11 | 7 | 3 | 9 | 5 | 1 | 7 | 3 | -1 | 10 | 6 | 2 | 8 | 4 | 0 | 6 | 2 | -2 |
| HHL | 6 | 2 | -2 | 4 | 0 | -4 | 2 | -2 | -6 | 5 | 1 | -3 | 3 | -1 | -5 | 1 | -3 | -7 | 4 | 0 | -4 | 2 | -2 | -6 | 0 | -4 | -8 |
| HHM | 10 | 6 | 2 | 8 | 4 | 0 | 6 | 2 | -2 | 9 | 5 | 1 | 7 | 3 | -1 | 5 | 1 | -3 | 8 | 4 | 0 | 6 | 2 | -2 | 4 | 0 | -4 |
| HHH | 14 | 10 | 6 | 12 | 8 | 4 | 10 | 6 | 2 | 13 | 9 | 5 | 11 | 7 | 3 | 9 | 5 | 1 | 12 | 8 | 4 | 10 | 6 | 2 | 8 | 4 | 0 |

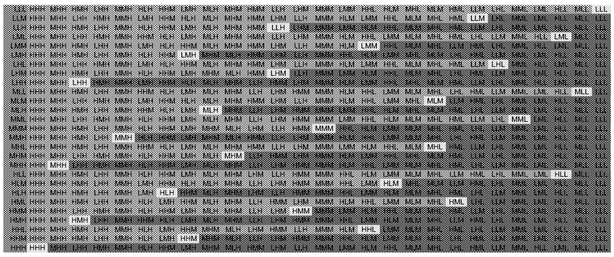
Figure 3: *Request-Receive* matrix displaying scores for requested and delivered QoS.

Figure 3 is an example of a *Request-Receive* matrix. The columns represent the possible QoS requirements of a data stream (high, medium or low in each of the three categories of performance, reliability and space efficiency), and the rows represent the possible attributes of the block groups of the VSS. The values in the matrix encode the match between the QoS required by the data stream and offered by locations on the VSS, where zero implies perfect match, positive numbers imply delivered QoS is better than requested QoS, and negative scores that delivered QoS is worse than requested QoS.

To calculate this matrix the user is asked to specify how much they value performance over reliability over space efficiency by choosing three integer constants $k_P$, $k_R$ and $k_S$ respectively. The rating for a particular combination of QoS requirements against those offered by the VSS is then calculated as:

$$k_P \times (P_{recv} - P_{req}) + k_R \times (R_{recv} - R_{req}) + k_S \times (S_{rece} - S_{req})$$

where $P_x$, $R_x$ and $S_x$ are 3 for high, 2 for medium and 1 for low. The constants used in Figure 3 were $k_P = 1$, $k_R = 2$ and $k_S = 4$.



| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| LLL | HHH | MHH | HMH | LHH | MMH | HLH | HHM | LMH | MLH | MHM | HMM | LLH | LHM | MMM | LMM | HHL | HLM | MHL | MLM | HML | LLM | LHL | MML | LML | HLL | MLL | LLL |
| LLM | HHH | MHH | LHH | HMH | MMH | LMH | HHM | HLH | MLH | MHM | HMM | LHM | LLH | MMM | HLM | LMM | HHL | MLM | MHL | HML | LLM | LHL | MML | LML | HLL | MLL | LLL |
| LLH | HHH | MHH | HMH | LHH | MMH | HLH | HHM | LMH | MLH | MHM | HMM | LLH | LHM | MMM | LMM | HLM | HHL | MLM | MHL | HML | LLM | LHL | MML | LML | HLL | MLL | LLL |
| LML | HHH | MHH | LHH | HMH | MMH | HHM | HLH | LMH | MLH | MHM | HMM | LLH | LHM | MMM | HLM | HHL | LMM | MLM | MHL | HML | LHL | LLM | MML | HLL | LML | MLL | LLL |
| LMM | HHH | MHH | HMH | LHH | MMH | LMH | HLH | HHM | MLH | MHM | HMM | LHM | LLH | MMM | HLM | LMM | HHL | MLM | MHL | LLM | HML | LHL | MML | HLL | LML | MLL | LLL |
| LMH | HHH | MHH | HMH | LHH | MMH | HLH | HHM | LMH | MHM | MLH | HMM | LHM | LLH | MMM | HHL | HLM | LMM | MHL | MLM | LHL | HML | LLM | MML | LML | HLL | MLL | LLL |
| LHL | HHH | MHH | LHH | HMH | MMH | LMH | HLH | HHM | MLH | MHM | HMM | LHM | LLH | MMM | HLM | LMM | HHL | MLM | MHL | HML | LLM | LHL | MML | HLL | LML | MLL | LLL |
| LHM | HHH | MHH | HMH | LHH | MMH | HLH | HHM | LMH | MHM | MLH | HMM | LHM | LLH | MMM | LMM | HLM | HHL | MLM | MHL | LHL | HML | LLM | MML | LML | HLL | MLL | LLL |
| LHH | HHH | MHH | LHH | HMH | MMH | LMH | HHM | HLH | MLH | MHM | LLH | HMM | LHM | MMM | HLM | LMM | HHL | MHL | MLM | HML | LLM | LHL | MML | HLL | LML | MLL | LLL |
| MLL | HHH | MHH | HMH | LHH | MMH | HHM | HLH | LMH | MLH | MHM | LLH | LHM | HMM | MMM | HLM | HHL | LMM | MLM | MHL | LHL | HML | LLM | MML | LML | HLL | MLL | LLL |
| MLM | HHH | MHH | LHH | HMH | MMH | LMH | HHM | HLH | MLH | MHM | HMM | LHM | LLH | MMM | HLM | HHL | LMM | MHL | MLM | LLM | HML | LHL | MML | LML | HLL | MLL | LLL |
| MLH | HHH | MHH | HMH | LHH | MMH | HHM | HLH | LMH | MLH | MHM | LLH | LHM | HMM | MMM | LMM | HHL | HLM | MHL | MLM | HML | LHL | LLM | MML | LML | HLL | MLL | LLL |
| MML | HHH | MHH | LHH | HMH | MMH | HLH | HHM | LMH | MLH | MHM | LLH | LHM | HMM | MMM | LMM | HHL | HLM | MLM | MHL | HML | LLM | LHL | MML | LML | HLL | MLL | LLL |
| MMM | HHH | MHH | HMH | LHH | MMH | HLH | HHM | LMH | MHM | MLH | LHM | LLH | HMM | MMM | HHL | HLM | LMM | MLM | MHL | HML | LHL | LLM | MML | LML | HLL | MLL | LLL |
| MMH | HHH | MHH | HMH | LHH | MMH | HLH | HHM | LMH | MHM | MLH | HMM | LLH | LHM | MMM | HLM | HHL | LMM | MLM | MHL | LHL | HML | LLM | MML | LML | HLL | MLL | LLL |
| MHL | HHH | MHH | HMH | LHH | MMH | HHM | HLH | LMH | MLH | MHM | HMM | LLH | LHM | MMM | LMM | HLM | HHL | MLM | MHL | HML | LLM | LHL | MML | LML | HLL | MLL | LLL |
| MHM | HHH | MHH | LHH | HMH | MMH | HLH | HHM | LMH | MLH | MHM | LLH | HMM | LHM | MMM | HLM | LMM | HHL | MHL | MLM | HML | LLM | LHL | MML | LML | HLL | MLL | LLL |
| MHH | HHH | MHH | LHH | HMH | MMH | HLH | HHM | LMH | MLH | MHM | LLH | LHM | HMM | MMM | HLM | HHL | LMM | MLM | MHL | HML | LHL | LLM | MML | LML | HLL | MLL | LLL |
| HLL | HHH | MHH | HMH | LHH | MMH | HLH | HHM | LMH | MLH | MHM | LHM | LLH | HMM | MMM | HHL | HLM | LMM | MLM | MHL | LLM | HML | LHL | MML | LML | HLL | MLL | LLL |
| HLM | HHH | MHH | HMH | LHH | MMH | LMH | HHM | HLH | MLH | MHM | LLH | LHM | HMM | MMM | HHL | LMM | HLM | MHL | MLM | LLM | HML | LHL | MML | LML | HLL | MLL | LLL |
| HLH | HHH | MHH | HMH | LHH | MMH | LMH | HLH | HHM | MLH | MHM | LHM | LLH | HMM | MMM | HHL | LMM | HLM | MLM | MHL | HML | LHL | LLM | MML | LML | HLL | MLL | LLL |
| HML | HHH | MHH | HMH | LHH | MMH | HLH | LMH | HHM | MLH | MHM | LLH | HMM | LHM | MMM | HLM | HHL | LMM | MLM | MHL | HML | LHL | LLM | MML | LML | HLL | MLL | LLL |
| HMM | HHH | MHH | LHH | HMH | MMH | HLH | HHM | LMH | MLH | MHM | LLH | LHM | HMM | MMM | LMM | HLM | HHL | MLM | MHL | HML | LLM | LHL | MML | LML | HLL | MLL | LLL |
| HMH | HHH | MHH | HMH | LHH | MMH | HHM | HLH | LMH | MLH | MHM | LLH | LHM | HMM | MMM | HHL | LMM | HLM | MLM | MHL | LLM | HML | LHL | MML | LML | HLL | MLL | LLL |
| HHL | HHH | MHH | HMH | LHH | MMH | HLH | LMH | HHM | MHM | MLH | LHM | HMM | LLH | MMM | HLM | HHL | LMM | MLM | MHL | HML | LHL | LLM | MML | LML | HLL | MLL | LLL |
| HHM | HHH | MHH | HMH | LHH | MMH | HLH | LMH | HHM | MHM | MLH | LLH | LHM | HMM | MMM | HHL | HLM | LMM | MLM | MHL | LHL | HML | LLM | MML | LML | HLL | MLL | LLL |
| HHH | HHH | MHH | LHH | HMH | MMH | HLH | HHM | LMH | MHM | MLH | HMM | LHM | LLH | MMM | LMM | HLM | HHL | MLM | MHL | LHL | HML | LLM | MML | HLL | LML | MLL | LLL |

Figure 4: Matrix showing the ordering of suitable storage locations against data stream QoS requirements.

We can then use this to form an ordering for matching the service levels of blocks on a VSS to a data stream's QoS requirements. For example, if a particular stream has the QoS requirements Low, Medium and High for performance, reliability and space efficiency (abbreviated to LMH in Figures 3 and 4), which VSS service levels would satisfy these and in what order? Figure 4 suggests an ordering of [HHH, MHH, HMH, LHH, MMH, HLH, HHM,....] in descending order of suitability.

This ordering is used by our simulator to intelligently place data. When a request is made by a stream to write data, the simulator traverses the address space of the VSS in the order specified by the column of the *Request-Receive* matrix which corresponds to the QoS requirements of the stream making that request until it finds sufficient free space to accommodate the data.

We also consider the migration of data from its initially allocated location. Such migration can occur in response to two situations: newly available capacity and changing stream characteristics.

**Newly Available Capacity** If the system gets to a high space utilisation, new data will most likely be placed in less satisfactory locations. However, should more satisfactory locations become available due to the deletion of data, data currently residing in less suitable locations can be migrated there.

**Changing Stream Characteristics** Migration is also required because streams are not static entities and their characteristics can change with time. Data may have been placed in one location according to the stream's initial access patterns or QoS requirements, but if these change it is necessary to move this data to areas which more closely match the new requirements.

## VISUALISATION

We have created a visualisation engine in Java for the simulation to show how the well the current data placement meets the user's QoS requirements. The visualisation consists of 3 main views: Stream View, Storage View and Graph View.

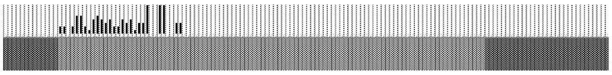


Figure 5: Stream View

Figure 5 shows how Stream View displays the placement and QoS requirement satisfaction of an individual data stream. The upper strip indicates where data currently resides, with a fully white bar meaning that a location is empty and a fully black bar that it is full. The lower strip represents how each area of the storage rates relative to the stream's requirements (Green – Better than required, Yellow – As required, Red – Underperforms).

Figure 6: Storage View

An additional panel known as the Storage View is available which visualises the storage as a whole (stream independent). Storage View is similar to an aggregated Stream View with the exception that is does not have the lower strip representing the storage system's offered QoS. This view is demonstrated in Figure 6.

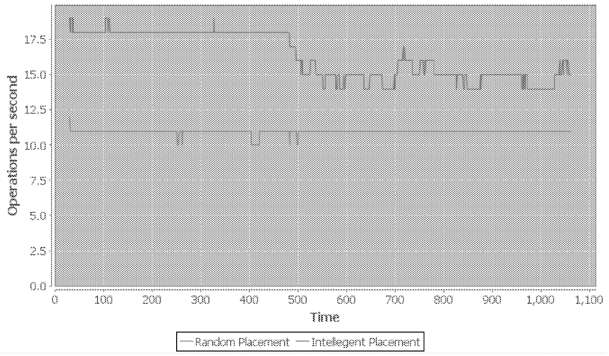


Figure 7: Graph View

Graph View is used to display the performance benefit of the intelligent placement technique, as shown in Figure 7. It displays in real-time a graph of the I/O operations per second achieved under the intelligent placement scheme, and also provides for comparison a graph of the performance of a random placement technique.

## EVALUATION

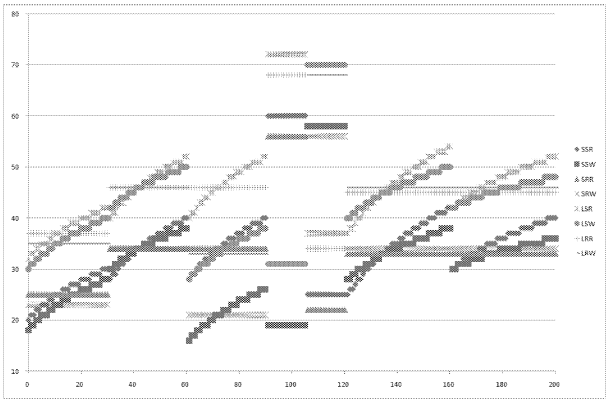


Figure 8: Graphical representation of the 8 profiles used for the case studies. The graph plots logical block address against throughput in bytes/second.

We demonstrate the applicability of our simulation through a number of case studies. Since the simulation requires an underlying profile of a VSS, we use a fabricated profile in order show some interesting characteristics of our techniques. This is shown in Figure 8, which plots logical address ($x$-axis) against throughput ($y$-axis), and is intended to simulate a VSS consisting of the following components:

1. Standard Magnetic Hard Disk

2. RAID 0 System

3. RAID 1 System

4. Read-Biased Flash Memory

5. Write-Biased Flash Memory

6. RAID 5 System

7. RAID 10 System

We will describe the characteristics of the streams used to drive the case studies using a 7-tuple of the form $\{[1,H,M,L,50\%,50\%,50\%]\}$. This example has an Arrival Rate of 1 I/O operation per second, High Performance requirements, Medium Reliability requirements and Low Space Efficiency requirements, with equal parts Small Sequential Reads to Large Random Writes. We use the QoS constants $k_P = 2$, $k_R = 3$ and $k_S = 1$ for all case studies. Videos of the case studies are at `http://www.doc.ic.ac.uk/~njd200/studies.html`.

### Case Study: Order of placement

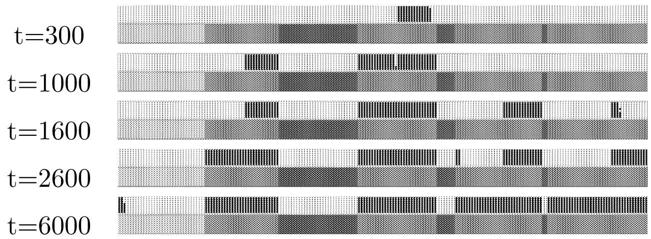


Figure 9: Sequential animation time-line showing correct placement ordering.

In order to verify the order in which the simulation places data, we executed the simulator with a single input data stream with the characteristics $\{[1,M,M,M,50\%,50\%,50\%]\}$. As shown in Figure 9, the green (better than required) portions of the storage are the first to be filled. Next (at $t = 6\,000$) the yellow (as required) portions of the storage start to be filled since there are no more green areas. In this example, since the simulation was only asked to fill 80% of the space, the red portions of the storage will never be used since no more space is needed.

### Case Study: Performance Comparison

The next experiment is to measure the performance of the placement method in terms of rate of I/O operations. In order to provide a basis for comparison we will also investigate the case where a random placement algorithm is employed. We explore two scenarios, both with a single input stream with characteristics {*[1,M,M,M,50%,50%,50%]*}:

- Scenario A: 80% free space aim
- Scenario B: 20% free space aim

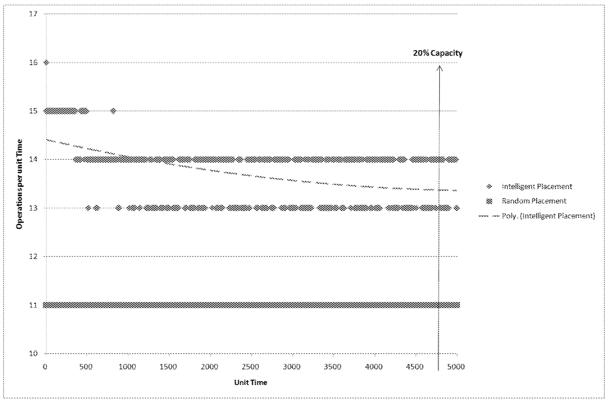


Figure 10: Performance of the VSS in Scenario A.

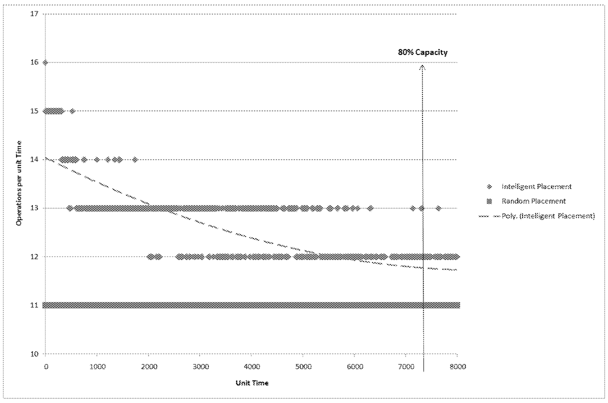


Figure 11: Performance of the VSS in Scenario B.

The graph for Scenario A (Figure 10) demonstrates that by using the intelligent placement technique over the random placement technique there is a performance increase of nearly 2.5 disk operation per unit time. Scenario B (Figure 11) only yields a performance increase of 0.75 disk operations per unit time. This is because when the system is kept at 20% of storage space in use, the probability of finding high performance areas of storage is high since they would not yet have been filled. This means that the majority of the data will be placed in these areas yielding very good performance for the system as a whole. When the percentage of storage space in use is set at 80%, the chance of finding a high performance area is far lower.

### Case Study: Multiple Streams

Since in reality many data streams will be competing with each other to allocate space, we now explore the case of multiple concurrent streams. Figure 12 shows the animation of a run with 3 streams: {*[1,H,M,L,17%,18%,84%]*, *[3,L,M,H,78%,81%,13%]*, *[1,M,M,M,50%,50%,50%]*}.

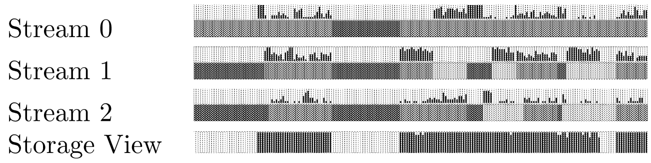


Figure 12: Multiple streams acting on one VSS.

Figure 12 demonstrates the multiple stream simulation. An interesting thing to notice about this run of the simulation is that, because each stream was given very different characteristics, the way each values its underlying storage is very different. There are certain areas of the storage that are preferred by all streams; this can lead to conflict between the streams.

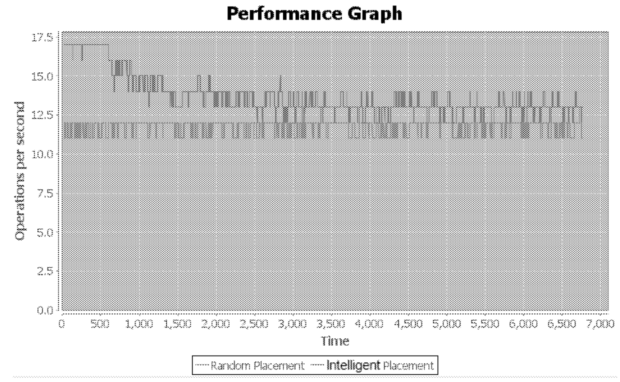


Figure 13: Performance profile when multiple streams are acting on the same storage.

In Figure 13 we observe that our intelligent placement scheme succeeds in allocating storage space such that performance is improved over results from a random allocation scheme.

### Case Study: Changing Stream Characteristics

We demonstrate the extent to which our simulation is capable of responding to changing access pattern characteristics by migrating data in a case study with a single stream with parameters {*[1,L,M,H,78%,81%,13%]*}.

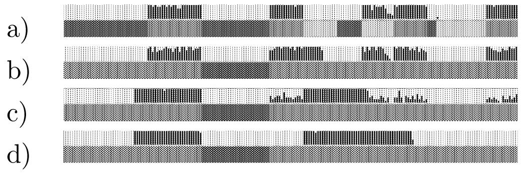


Figure 14: Changing QoS stream characteristics.

Figure 14 shows an example of a simulation where there is a change in the stream's characteristics mid run. The four snapshots correspond to the follow situations:

**a):** Shortly before stream characteristics change.

**b):** The point at which the stream's characteristics change. Note that portions of the storage that were unsuitable are now classified as good.

**c):** The simulation in mid-migration.

**d):** The final resting place of the data post-migration.

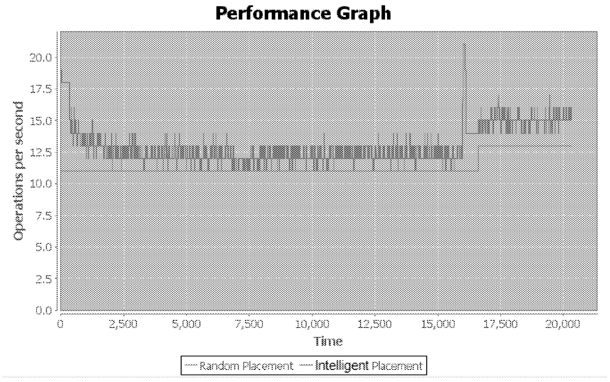


Figure 15: Performance where a stream's characteristics have been changed in mid-simulation.

Figure 15 shows a performance graph of this run of the simulation. The most noticeable aspect of the graph is the spike at $t = 16\,000$, caused by the migration process writing the poorly placed data to the best available space, which is very high performance. Also note the system's performance is higher post migration.

### Case Study: Changing Space Consumption

It is to be expected that with time a system's space utilisation will grow and shrink. In this context we investigate the extent to which migration can be applied successfully. Figure 16 shows an example of the simulation with a single stream with settings {*[1,M,M,M,50%,50%,50%]*}.

**a):** The file system reaches its highest space utilisation.

**b):** The file system at its lowest space utilisation.

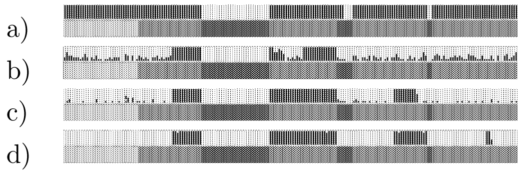


Figure 16: Example of a simulation when the file system grows and then shrinks.

**c):** Migration is 50% complete.

**d):** The post-migration layout of the data.

Once again, performance is higher post migration.

## CONCLUSION

This paper has described methods of profiling and characterising a storage device under different conditions to achieve an in-depth picture of how the device will perform. We have then discussed the placement and migration of data on such a storage device to best meet users' performance, reliability and space efficiency requirements. This has been implemented in a simulation where data streams can be specified to operate on the storage, and we have demonstrated this in operation through a number of case studies.

There are many opportunities for further work. One attractive feature of VSSs is the ability to add new devices dynamically. This requires a mechanism for detecting when such an event has occurred and automatically re-profiling the VSS. We have also not modelled the failure of hard drives and the operation of RAID systems in degraded mode, but these would obviously have a major effect on performance and would complicate placement and migration. Finally, real-life processes often anticipate the amount of storage they will require and provisionally pre-allocate space to ensure that their data is not fragmented, but this is not currently supported.

## REFERENCES

M. Bhadkamkar, J. Guerra, L. Useche, S. Burnett, J. Liptak, R. Rangaswami, and V. Hristidis. BORG: Block-reORGanization for Self-Optimizing Storage Systems. In *Proc. 7th USENIX Conference on File and Storage Technologies (FAST'09)*, pages 183–196, San Fransisco, CA, February 2009.

F. Franciosi and W.J. Knottenbelt. Towards a QoS-aware virtualised storage system. In *Proc. 25th UK Performance Engineering Workshop (UKPEW'09)*, pages 49–60, Leeds, July 2009.

J. Rydning and D. Reinsel. Worldwide hard disk drive 2009–2012 forecast: Navigating the transitions for enterprise applications. IDC Market Analysis, Document 216394, February 2009.

# DATAPATH ARCHITECTURE SIMULATION [*]

Venkatesh Kannan,
Marc Voorhoeve
Dept. of Mathematics and Computer Science
Eindhoven University of Technology
E-mail: {v.kannan,m.voorhoeve}@tue.nl

Lou Somers
Océ Research and Development, the Netherlands,
Dept. of Mathematics and Computer Science
Eindhoven University of Technology
E-mail: lou.somers@oce.com

[*] Research carried out in the context of the Octopus project under the responsibility of the Embedded Systems Institute. This project is partially supported by the Netherlands Ministry of Economic Affairs under the Senter TS program.

## KEYWORDS

Petri Nets, CPN Tools, Scheduling.

## ABSTRACT

In designing embedded systems, such as printers, it is vital to select the proper subcomponents and to fully realize their potential. In this paper, we propose a simulation-based approach to assess and improve tentative printer datapath architectures.

## INTRODUCTION

Present-day printers process all kinds of documents, involving both paper-based and data-only formats. An important part is the digital datapath, that transforms the input data (PostScript or scanned image) to the output data format. The datapath functionality warrants the coordination of various resources, allowing their concurrent use. Developers face the task of establishing architectures by selecting and connecting resources, as well as specifying scheduling and resource allocation strategies. Architecture proposals need to be assessed by evaluating their performance. We will sketch a simulation-based approach to achieve this, concentrating on the datapath architecture and abstracting from the physical paper-and-ink aspects. Our research is an elaboration of (Igna et al. 2008).

A printer system contains a scanner as input device, a printer (in the narrow sense) as output device, whereas a network port serves as input-output device. Between input and output, a series of image processing tasks are performed, depending on the nature of the requested job type (e.g. print, scan, copy). In (Brassé and de Smet 2008), typical examples of image processing are given. Steps are executed concurrently, requiring resources such as memory for temporary storage (e.g. volatile RAM, flash memory, hard disk), buses for data transport, and processing hardware (e.g. general purpose CPU, graphical processor, FPGA block, ASIC).

Typically, in a printer datapath, the requested image quality may not be sacrificed for speed. An end user may observe that the printer slows down when a complex image is processed. In a video environment, this requirement is reversed; image quality must be sacrificed, if necessary, in order to achieve the timely delivery of the images.

In designing a datapath architecture, the resources should be balanced in order to meet the performance and quality requirements without unnecessary costs. Simulation is used to assess the consequences of given architectures and scheduling policies, producing charts and tables that allow to identify performance bottlenecks and thereby suggest improvements. Given the concurrent nature of the modeled system and the need for hierarchy, we use CPN Tools (http://wiki.daimi.au.dk/cpntools/cpntools.wiki; Jensen 1992). High-level Petri nets like CPN are often used for performance modeling and evaluation, viz. (Frigerio and Salice 2008; Machado et al. 1998; Sokolov et al. 2007).

Jobs are characterized by their use case (e.g. color-copying with 141% zoom) and parameters (e.g. input and output paper size). Figure 1 depicts an example use case, defining the job structure of a class of simple copy jobs. It consists of scanning, image processing, and then printing (in several copies) a document consisting of multiple pages. An actual job belongs to a use case and possesses several job parameters, such as the number of copies. In the depicted use case, each document page is scanned, followed by resampling, halftoning, and finally printing. The printed images are also uploaded; these uploaded images must be downloaded in order to print additional copies.

The use case in Figure 1 contains the parameters $d$ (document page) and $c$ (copy number). The function $p(d)$ gives the number of page $d$; initially, because of the initial token in place $x$, the start scan event takes the page $d$ satisfying $p(d) = 1$ (the first page). When resampling has finished, the page number is incremented. The pair $(d, c)$ consists of page $d$ and a copy number $c$. The function $nxt(d, c)$ increments the page number, so $nxt(d, c) = (p(d) + 1, c)$ unless $d$ is the last page, in which case $nxt(d, c) = (1, c + 1)$. If $d$ is the last page and $c$ the last copy, then $nxt(d, c)$ is empty. Because of the initial token in the place $y$, downloading starts from the second copy onward.

Dependencies between tasks are indicated in CPN notation. Each task consists of a start and end event and may be parameterized. For instance, because of the ini-

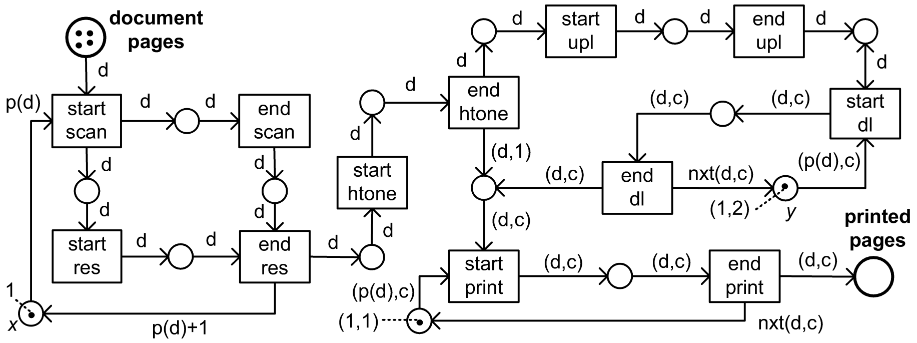


Figure 1: Use case partial order

tial token in place $x$, the start-scan event in the figure is enabled with a document page $d$ satisfying $p(d) = 1$. As the function $p(d)$ gives the page number of a document, this ensures that scanning is started with the first page. Resampling is pipelined with scanning, so it can start before scanning has ended. When resampling has ended, the next page (with page number $p(d) + 1$) can be scanned concurrently with halftoning the current page. Halftoning needs the full image, so it is not pipelined. Uploading and printing the first copy of the halftoned images are concurrent. For printing copies beyond the first, the images must be downloaded, which is ensured by the initial token in place $y$.

Figure 2 gives in part two possible execution charts of the job of Figure 1, for an architecture with a single bus resource that is fully occupied by uploading and downloading jobs. The printing of the first copies $p(1,1)..p(4,1)$ ($p(1,1)$ occurs between start print and end print with $d = 1, c = 1$) and the uploading $u(1)..u(4)$ are concurrent. Uploading and downloading needs the same resource, so concurrency between these tasks is not realizable. In the topmost chart, the uploading of the first page $u(4)$ causes a delay in the printing of the second copy $p(1,2)$, because $d(1,2)$ (downloading page 2) has to wait until $u(4)$ has terminated. A different scheduling policy that allows preemption of the $u(4)$ task may reduce the printer idle time, as shown in the bottom chart.

In the sequel, we describe the models and experiments of a datapath simulation study that we have carried out.

## MODELING

### Top level and job handling

Modeling with CPN tools (and similar high-level Petri nets) presents a choice between adding structure to the net model or to the token colors. Typically, a balance is maintained between the two. In our model, a division is made between job and resource handling. The job handler consists of use cases depicted as partial orders like Figure 1; so the net structure represents the job workflow. The resource handler instead uses data (token colors) to control the flow. Resources are not modeled individually, as, unlike use cases, many possible architecture choices are possible. Representing these choices by colors rather than by net structure offers more flexibility.

The job handler controls the workflow, keeping track of enabled events, which are transferred to the resource handler through the request place. The requests contain all relevant parameters for the resource handler, such as task sizes. Depending on resource availability, the resource handler takes care of executing the event and enabling successor events after appropriate timing delays. These delays are computed from the resource characteristics and the event parameters. The successor triggers are communicated back to the job handler. Note that a single event may trigger several successors, viz. start scan in Figure 1. The modeling closely resembles the nets-as-tokens approach from (Valk 1998); the use case partial orders can be interpreted as net tokens handled by the resource handler substitution transition

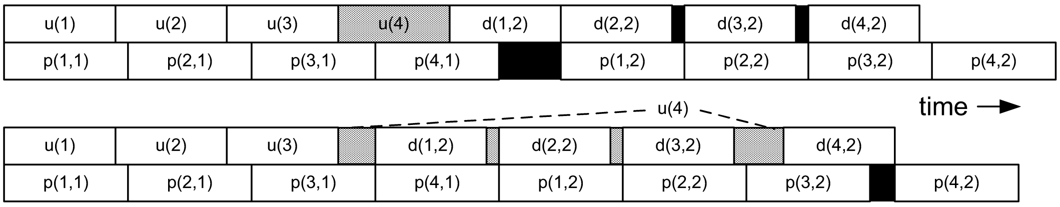


Figure 2: Gantt charts for bus and printer resources

(subnet).

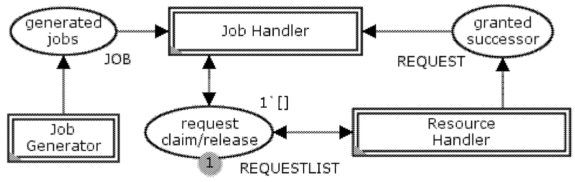


Figure 3: Top-level Model

Figure 3 depicts the top level of our simulation model. The job handler contains a partial order subnet for each use case. Figure 1 is an example subnet; the actual CPN models contain more image processing events than depicted in Figure 1. Also, additional interface arcs allow events to communicate with the resource handler. The job generator can either generate a fixed set of print jobs or a random stream that might occur in practice.

### Resource handling

The Resource Handler subnet in Figure 4 deals with individual events and contains information about the system architecture, resource allocation and scheduling strategies. It maintains information about the occupation of resources and the tasks that occupy them. Note that some tasks allow preemption - they may be interrupted and their resources redistributed. Some resources, like data buses, can be partially allocated to tasks, allowing them to proceed at a slower rate.

Some information about the successors of requested events is needed in addition to the resource state. Currently, a data structure containing use case information is provided.

Figure 4 depicts the Resource Handler subnet. All requests submitted to release resources are handled by the release resources transition, recalculating the resource capacities stored in the resource state and moving the event requests from the request claim/release place to the requests place. The requests place contains a queue of events waiting for their requested resources to become available to them, based on the resource allocation rules and scheduling policy implemented. Once the resources for a waiting event are allocated, the handle request transition updates the resource state and adds the successor events, determined from the partial order data structure, to the running jobs (busy tasks) list. This list contains records consisting of job information, event name, remaining size and current rate. If the current rate of a busy task is not modified, its time to completion equals the remaining size divided by the current rate. Note that the reallocation of resources can modify the current rate.

The subnet task manager is responsible for monitoring the running jobs. This subnet contains timed color sets; it computes the required delays. Whenever a new successor event is added to the running jobs list, a trigger is created by adding a token in the trigger place. This enables the task manager, which calculates the earliest finish time (EFT) for the jobs in the running jobs list based on their remaining size and current rate attributes. If there are no further new additions to the running jobs list, the task manager removes the event with the EFT equal to the current model time and sends a trigger to enable the corresponding event in the Job Handler partial order through the granted successor interface place. Additionally, the task manager recomputes the remaining size, current rate and EFT for the remaining entries in the running jobs list. Note that a calculated EFT can either be invalidated by a new addition to the running jobs list with a smaller completion time, or by influencing the current rate of entries in the running jobs list. In some cases, this causes superfluous recalculation of the remaining size and current rate of the entries in the running jobs list. This is a result of the constraint in CPN Tools that a token with a given time stamp cannot be removed or updated before its due time.

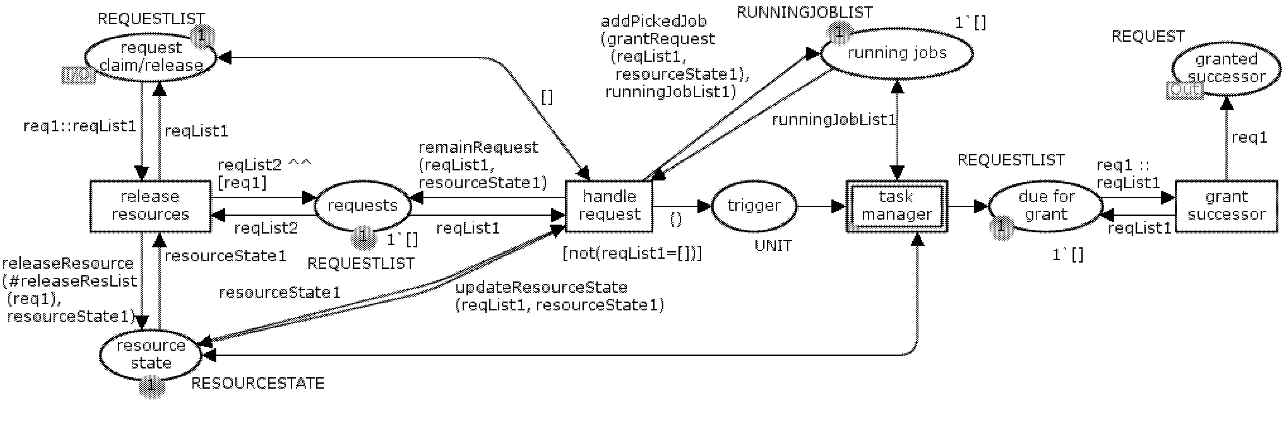


Figure 4: Resource Handler

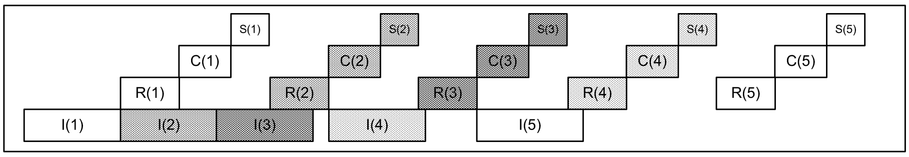


Figure 5: Gantt task chart for strategy BS-21.

The sketched division between job and resource handling greatly improves the clarity and reusability. There is a price attached, though. The resource handler, which is responsible for scheduling decisions, has less information about tasks to be executed, which makes lookahead scheduling harder to implement.

## SIMULATION EXPERIMENTS

Simulation models have been built for a use cases that differs somewhat from Figure 1. These models have been compared to an implemented prototype for validation purposes. In addition, Gantt charts produced by the simulation were inspected by the prototype engineers. The validation led to a reduction in expected size of input documents (reducing durations and resource usage). Also, the duration distribution for the uploading and decompression tasks that precede printing in the use case had to be adapted. Then, new printer datapath architectures were modeled and experimented with, resulting in various proposals for improvement. One example addresses the buffers $X, Y$ used for storing intermediate results of the image processing tasks $I$ (interpreting), $R$ (rendering), $C$ (compressing), and $S$ (storing). These tasks occur in many print use cases and must be executed in sequence for any page. Step $I$ claims/writes to $X$; task $R$ reads from/releases $X$ and claims/writes to $Y$; task $C$ overwrites $Y$; task $S$ reads from/releases $Y$. The average durations for the $I, R, C$ and $S$ tasks were respectively set to 1.61, 0.99, 0.95 and 0.67 time units. Buffer capacity is claimed at the start of a task and released upon termination. The CPN model for this experiment consist of about 100 nodes (places and transitions), 50 functions and 30 data types (color sets) in total. The job handler subnet contains the majority of the nodes, that were of a very generic nature. The resource handler is responsible for most arc inscriptions and function definitions. Monitors were added for recording execution times and resource utilization. Explorative simulations were carried out, simulating the processing of 500 pages within several architecture choices.

Figure 5 symbolically charts a simulated run containing a five-page print job where buffers $X, Y$ allow respectively two and one images to be stored concurrently. This buffer strategy (BS) is called BS-21. The charted run clearly shows gaps (idle time) for each of the four tasks. By increasing buffer $Y$, these gaps can be reduced.

The chart in Figure 5 has been simplified; due to variations in page contents and CPU utilization, the duration of tasks is not constant. In the actual model, stochastic variables are used. Choosing distributions for the task durations did require focussed breadboard experiments. Moreover, shortage of non-buffer resources may cause additional gaps. The actual experiments did take these influences into account.

After an exploratory phase, a few promising designs were selected for an in-depth simulation study processing 30 subruns with 500 pages, taking about 25 minutes of simulation time on an Intel Centrino Duo 2.20 GHz CPU with 2GB RAM per alternative. These simulations resulted in performance characteristics with confidence intervals (given as half widths) for the chosen alternatives. Table 1 lists five buffer strategies BS-$xy$, with buffers $X, Y$ allowing the storage of respectively $x, y$ images.

In the simulation experiments, jobs amounting to 500 pages were generated for each modeled architecture. Job generation was then stopped and the systems ran to completion. The task durations were randomized with experimentally determined normal distributions. The simulated metrics include duration until completion (cdur) with 95% half width (hw). Also, the idle times of task $I$ (it$I$) with half widths are listed and likewise for tasks $R$, $C$, and $S$.

Table 1 shows that task $I$ has the least idle time, so it is probably a bottleneck. During the execution of a job, a scheduler should keep idling of bottleneck resources to a minimum. Architectures BS-11, BS-21, BS-12 perform suboptimally in this aspect. A dramatic improved is achieved by BS-22. A further increase of buffer sizes gives only marginal improvements. The confidence intervals clearly indicate that these deductions are valid. Note that by increasing the efficiency (strategies BS-22 and BS-23), the standard deviation becomes larger, as running jobs can influence one another in more ways.

We have extended the above experiments, involving the bandwidth of the bus used by the storage task, which influences the duration of that task. Figure 6 plots the duration until completion with respect to the bandwidth for each of the five buffer strategies.

Note that strategy BS-12 does not benefit from a bandwidth increase. This illustrates the fact that task $I$ is critical for this strategy. Reducing the duration of storage thus will not affect the overall duration.

Table 1: Buffer strategy simulation results

| arch. | cdur | hw | it$I$ | hw | it$R$ | hw | it$C$ | hw | it$S$ | hw |
|---|---|---|---|---|---|---|---|---|---|---|
| BS-11 | 1373 | 2.3 | 572 | 1.6 | 876 | 1.8 | 924 | 2.1 | 1032 | 2.2 |
| BS-12 | 1292 | 2.6 | 492 | 1.3 | 798 | 1.8 | 844 | 3.2 | 943 | 3.3 |
| BS-21 | 1279 | 2.7 | 472 | 3.0 | 781 | 2.6 | 830 | 1.5 | 945 | 1.7 |
| BS-22 | 847 | 5.7 | 52 | 8.4 | 354 | 8.2 | 405 | 8.7 | 495 | 9.7 |
| BS-23 | 842 | 6.7 | 47 | 9.4 | 349 | 9.5 | 401 | 9.7 | 489 | 10.4 |

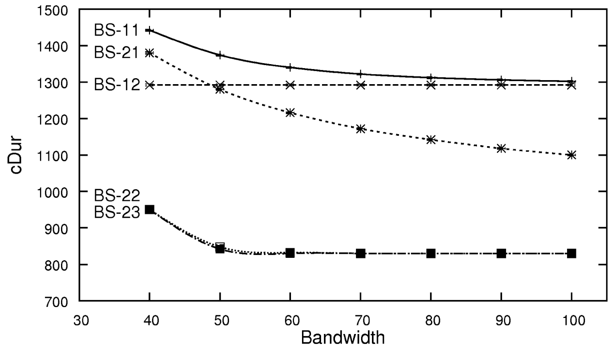


Figure 6: Buffer Strategies and Bandwith Durations

## CONCLUSION

There is much literature on comparing distributed system architectures (e.g. flexible manufacturing systems) by simulation. In this vein, we have studied printer datapaths. Our approach allowed to compare tentative designs and measure their performance under more or less realistic circumstances. Experimenting with simulation models - in contrast to hardware prototypes - allows full access to the state, giving much more opportunities to detect possible causes of suboptimal behavior. Of course, there are costs involved in modeling (e.g. to obtain duration distributions) and validating the models, but the costs of, e.g., developing an emulator are much higher.

Using colored Petri nets for modeling the use cases as in Figure 1 allows runtime inspection of states from the tokens representing busy or waiting tasks. Thus, queues caused by resource shortage can be observed, which is a great help in diagnosing the model's performance. Exploration can even be improved by allowing models to be adapted at run-time, saving time that would have been needed to modify, recompile and restart. CPN Tools allows to explore alternatives, while retaining the possibilities to replicate experiments and compute confidence intervals for in-depth simulation experiments with reliable conclusions.

Simulation methods do supplement the methods addressing the steady state behavior for jobs consisting of "infinitely" many pages. Such methods allow the search for optimal designs in certain circumstances (Ghamarian et al. 2006). However, for a queue containing several finite jobs, a schedule producing good steady state behavior for each individual job may perform not so well. Assessing the schedules by simulation is therefore necessary.

The sketched approach presents a few challenges, such as calibrating the needed parameters as well as the knowledge transfer needed between the developers of the system architecture and of the simulation model. Ideally, the engineers involved in printer development should also create the simulation models and conduct the experiments. We plan to support developers without extensive knowledge of CPN Tools by a toolkit that generates simulation models from a specification of the design choices.

## REFERENCES

Brassé, M. and S. de Smet 2008 "Data path design and image quality aspects of the next generation multifunctional printer." *Proceedings SPIE Image Quality and System Performance*, SPIE 2008.

Frigerio, L. and F. Salice 2008 "A performance-oriented hardware/software partitioning for datapath applications." *Proceedings CODES+ISSS'08*, ACM.

Ghamarian, A.; M. Geilen; S. Stuijk; T. Basten; A. Moonen; M. Bekooij; B. Theelen; and M. Mousavi 2006 "Throughput analysis of synchronous data flow graphs." *Proceedings ACSD'06*, IEEE.

Igna, G.; V. Kannan; Y. Yang; T. Basten; M. Geilen; F. Vaandrager; M. Voorhoeve; S. de Smet; and L. Somers 2008 "Formal Modeling and Scheduling of Data Paths of Digital Document Printers." *Proceedings Formats'08*, Springer.

Jensen, K. 1992 "Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use." *EATCS monographs on Theoretical Computer Science*, Springer 1992.

Monahan, C. and F. Brewer 1995 "Symbolic Modeling and Evaluation of Data Paths." *Proceedings DAC'95*, ACM.

Machado, R.; J. Fernandes; A. Proença; and J. Fern 1998 "An Object-Oriented Model for Rapid Prototyping of Data Path/Control Systems - A Case Study." *Proceedings INCOM'98*, Pergamon.

Sokolov, D.; I. Poliakov; and A. Yakovlev 2007 "Asynchronous Data Path Models." *Proceedings ASCD'07*, IEEE.

Valk 1998 "Petri Nets as Token Objects." *Proceedings ATPN'98*, Springer.

# State Estimation of a Nonlinear CSTR Using a Novel Asynchronous Data Fusion Based on Adaptive Extended Kalman Filter

Vahid Fathabadi
Mehdi Shahbazian
Karim Salahshoor
Lotfollah Jargani
Petroleum University Of Technology
Department of Instrumentation and Automation ,
P.O.Box 12333,Tehran,
Iran
E-mails: vahidfathabadi@gmail.com, shahbazian_m@yahoo.com, salahshoor@put.ac.ir, pooria_jargani@yahoo.com

**KEYWORDS**
Multi sensor fusion systems, Decentralized data fusion, Extended Kalman filter, Adaptive Fading Extended Kalman filter, State estimation.

## ABSTRACT

This paper presents the state estimation problem for nonlinear industrial systems using asynchronous measurements to simulate the circumstances of practical processes. There exist three distinct difficulties encountered in real-world applications, i.e. lack of perfect knowledge on model-reality mismatch and noise distribution matrices, diversified sampling rate and communication delays. For the first drawback, an adaptive fading extended Kalman filter (AFEKF) is utilized to simultaneously alleviate both model uncertainty and measurement noises. For the second problem, a distributed AFEKF is proposed to cover the issue of multi-rate measurement signals. To meet the last challenge, three methods are proposed which encompass the fusion of modified AFEKF and Alexander EKF(Alexander 1991) method. A comparative study was further conducted on a simulated nonlinear CSTR to demonstrate the extent of improvement achieved over the existing methodologies. Simulation outcomes indicate a significant superiority of the proposed approaches.

## INTRODUCTION

Data fusion is the process of dealing with association, correlation and combination of data and information from single or multiple sources to achieve refined position and identity estimates. Multi-sensor fusion systems (MFS) often receive data from dissimilar sensors. Hereafter, the system should consider the coordination between sensors such as different communications or inherent delays and also different sampling rate.

In this paper, the nontrivial problem of multi-sensor multi-rate data fusion is reformulated in a distinct manner to let for a more specific treatment. Three sources of difficulties can be distinguished accordingly; these are: 1) Model-reality mismatch and unknown noise distribution matrices. 2) Measurement signals with different sampling rates. 3) Different communication delays.

Among the various techniques available for multi-sensor data fusion, Kalman filtering-based approach is one of the most significant one, as it proves to be an efficient recursive algorithm suitable for real-time applications. In classical Kalman filter approaches, however, a Gaussian distribution is assumed *a priori* for both process and measurement noise signals; a centralized fusion filter that assumes all observation coming with known fixed specifications, i.e. mean and covariance matrices. For many real-world applications this may not truly be the case. Some methods have been proposed in literature to meet this challenge. In (Authur et al. 1987), the adaptive Kalman filter banks for weights have been used to compute covariance. In (Moose 1975), the output correlation has been used to compute Kalman gain without care of this covariance. In this paper an AFEKF approach is proposed to reduce the effect of model-uncertainty and inaccurate assumptions made on noise characteristics.

Asynchronous data fusion is used when measured signals are received with different sampling rate and communication delays. In the time delays context, a common approach is the partial differential equation (PDE) approach used frequently in literature. This approach is usually related to solving a partial differential equation with boundary condition which do not have an explicit solution in general. For the case of discrete-time systems, the problem has been investigated via system augmentation and standard Kalman filtering. Note that the augmented Kalman filtering approach is computationally expensive, especially when the dimension of the system is high and the measurement lags are large. On the other hand, the polynomial approach only addresses the steady-state filtering problem and it requires solving a much higher order of spectral factorization for systems with delays. To this end, we here present novel technique which sum up the advantage of AFEKF (Authur et al. 2009) and Alexander method (Alexander 1991). A recalculation methodology as a well-established approach is also examined to probe its viability within the new framework. A careful comparative study was carried out via a simulated nonlinear CSTR to illustrate the effectiveness of the proposed algorithm in this paper.

The rest of this paper is organized as follows: Based on non-linear process and incomplete noise statistical character, state estimation procedure based on EKF and adaptive fading methods has been derived in following section. The next section shows how the EKF algorithm can be changed to accommodate latency in the measurements due to both communication delay and multi-rate sensors. In next section, CSTR industrial plant will be simulated. Simulation results illustrating the performance of the proposed state estimation approaches are presented in next section. Finally, last section summarizes the main conclusions.

## PROPOSED METHODOLOGY

### Extended Kalman Filter

Extended Kalman filter (EKF) gives a simple and effective remedy to overcome nonlinear estimation problem. Its basic idea is to locally linearize the nonlinear functions at each sampling time instant around the most recent process condition estimate.

$$x(k) = F(k)x(k-1) + B(k)u(k-1) + w(k-1) \quad (1)$$
$$z(k) = H(k)x(k) + v(k) \quad (2)$$

Where the state transition matrix F(k), the input matrix B(k), and the observation matrix H(k) are the Jacobian matrices which are evaluated at the most recent process operating condition in real-time rather than the process fixed nominal values. Now, a linear state space system and a linear measurement equation are in (1) and (2) respectively. That means standard Kalman filter equations can be used to estimate the state. Thus, the following equations are named as the EKF equations:

$$\hat{x}_k^- = f_{k-1}(\hat{x}_{k-1}^+, u_{k-1}) \quad (3)$$
$$P_k^- = F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \quad (4)$$
$$K_k = P_k^- H_k^T\left(H_k P_k^- H_k^T + R_k\right)^{-1} \quad (5)$$
$$\hat{x}_k^+ = \hat{x}_k^- + K_k(Z_k - h_k(\hat{x}_k^-)) \quad (6)$$
$$P_k^+ = (I - K_k H_k)P_k^- \quad (7)$$

### Adaptive Fading Extended Kalman Filter

The extended Kalman filter formulation assumes complete a priori knowledge of the process and measurement noise covariance matrices $Q_k$ and $R_k$. However, in most practical applications these matrices are initially estimated or, in fact, are unknown. The problem here is that the optimality of the estimation algorithm in the extended Kalman filter setting is closely connected to the quality of the *a priori* noise statistics. It has been shown how poor estimates of the input noise statistics may seriously degrade the Kalman filter performance, and even provokes the divergence of the filter (Fitzgerald 1971). From this point of view it can be expected that an adaptive formulation of the extended Kalman filter will result in a better performance or will prevent filter divergence.

In this case, the covariance of the adaptive fading algorithm is(Authur et al. 2009):

$$C_k = E\left[\eta_k \eta_k^T\right] = H_k P_k^- H_k^T + R_k \quad (8)$$

Where $\eta_k = z_k - h_k(\hat{x}_k^-)$, $P_k^-$ and $R_k$ are innovations, a predicted error covariance and a measurement covariance of the EKF, respectively. $C_k$ has been referred as the calculated innovation covariance. In general, the innovation of the filter is easily affected by unaccounted errors, such as an unknown fault bias, an un-modeled dynamic, or an unknown initial condition. Also, the innovation covariance shows the effect of any unaccounted errors, as they are directly involved in the computations of the innovation.

As a result, the change of an innovation covariance can be used for an adaptive filter. The increased innovation covariance can be estimated as

$$\bar{C}_k = \frac{1}{M-1}\sum_{i=k-M+1}^{k} \eta_i \eta_i^T \quad (9)$$

Where M is a window size. $\bar{C}_k$ is called the estimated innovation covariance in this paper.

To account for the effect of the unaccounted system model errors, Kim (Kim 2006) proposed the AFEKF. In this section, we summarize the structure of the AFEKF. Assume that the exact dynamic or measurement equation is unknown. The relation between $C_k$ and $\bar{C}_k$ is defined as $\bar{C}_k = \alpha_k C_k$. Then, the scalar variable $\alpha_k$ can be estimated by

$$\alpha_k = \max\left\{1, \frac{1}{m}\mathrm{tr}(\bar{C}_k C_k^{-1})\right\} \quad (10)$$

Where m is the dimension of $z_k$, which is the m×1 measurement vector. When the innovation covariance is increased by unaccounted errors, an estimated innovation covariance $\bar{C}_k$ shows the estimate of the true innovation covariance.

We consider the first case, in which the dynamic equation is not known exactly. Generally, the effects of incomplete information in the dynamic equation can be compensated by the increase of the magnitude of $P_k^-$ . Thus a predicted error covariance must be increased to compensate the effect of an inexact dynamic equation as $\bar{P}_k^-$ where $\bar{P}_k^- = \lambda_k P_k^-$. Here $\lambda_k$ is called a forgetting factor and $\lambda_k \geq 1$. Then $\bar{C}_k$ can be represented by

$$\bar{C}_k = H_k \bar{P}_k^- H_k^T + R_k = H_k(\lambda_k P_k^-)H_k^T + R_k \quad (11)$$

In (11), we can obtain the following equations

$$\alpha_k\left[H_k P_k^- H_k^T + R_k\right] = \lambda_k H_k P_k^- H_k^T + R_k$$
$$\lambda_k \approx \frac{\mathrm{tr}(\alpha_k H_k P_k^- H_k^T + (\alpha_k - 1)R_k)}{\mathrm{tr}(H_k P_k^- H_k^T)} \quad (12)$$

Here, (12) gives an approximate value of $\lambda_k$. However, the measurement equation does not have unaccounted errors in the first case. As a matter of fact, an innovation covariance is increased by the increased predicted error covariance, not the measurement covariance. This indicates that the ratio of innovation covariances $\alpha_k$ is mainly generated by $\lambda_k$.

Therefore it can be assumed that $\alpha_k$ is almost equal to $\lambda_k$. With the assumption of $\lambda_k = \alpha_k$, the error covariance is $\bar{P}_k^- = \alpha_k P_k^-$. The AFEKF using this concept is denoted as "the AFEKF with rescaling-$P_k$ ."

Next, we consider the second case, in which the measurement equation is not known exactly. The estimation error and the innovation covariance may be also increased by the effect of the unknown information, as they were in the first case. Here, the dynamic equation does not have unaccounted errors in the second case. So, an innovation covariance is increased by an increased measurement covariance, not a predicted error covariance. The effects of incomplete information in the measurement equation can be compensated by the decrease of the magnitude of $K_k$. We set $\lambda_k$=1 because the predicted error covariance is unchanged as $\overline{P}_k^- = P_k^-$ and we use the Kalman gain that is decreased by $1/\alpha_k$. The decrease of the Kalman gain magnitude means that it depends less on measurement information. The AFEKF using this concept is denoted as "the AFEKF with rescaling-$K_k$."

## ASYNCHRONOUS KALMAN FILTER

In the previous section, it was assumed that all the sensor measurements are synchronously available at each sampling instant. This unrealistic assumption must be disregarded in according to the communication delay or the different sensor sampling rates that affect the multi-sensor data fusion procedure.

### Asynchronous Communication Delay

A nonlinear discrete system observed by non-delayed measurements where both process and measurements are influenced by additive Gaussian noise can be put in state space form in (1) and (2).

Furthermore, if this system has an output that is delayed n samples, for instance due to a slow sensor or a long processing time of the sensor data, there will be a second output equation (Authur et al. 1998):

$$z_k^* = h^*(x_s, s) + v_k^* \qquad (13)$$

Where

$$s = k - N$$

The delayed measurement cannot be fused using the normal extended Kalman filter equations but requires some modifications in the structure of the filter.

*Recalculation Method*

If only a few measurements are fused in the delay period or if the computational burden of the filter is uncritical, an optimal filter estimate can be obtained simply by recalculating the filter through the delay period. As the measurement are not available in the time interval t=S to t=K, it is suggested to update state and covariance without measurement update in this time interval. As soon as measurement of time t=S is received with delay at t=K, estimation procedure begins with time update and measurement update again from t=S, and will be proceed to time t=K+1 using only time update equations. The repeated manner of this procedure imposes high computational burden. Thus the following methods are proposed to overcome this drawback.

*Updating State at different times(Alexander method)*

Using the standard extended Kalman filter equations, the measurement $z^*(k)$ should be fused at time s, causing a correction in the state estimate and a decrease in the state covariance. As the state covariance matrix decides the Kalman gain, the measurements occurring after this will all be fused differently than if the measurement update for z*(k) is omitted. Therefore, if the measurement z*(k) is delayed n samples and fused at time k, the update data should reflect the fact that the n data updates from time s to k. Therefore, the state and covariance estimates have all been affected by the delay in a complex manner.

Equations that account for this when fusing z*(k) at time k has been derived in (Alexander 1991) but are of such complexity that they are not feasible in many cases. It is, therefore, suggested that if the measurement sensitivity matrix H*(s) and the noise distribution matrix, R*(k) are known at time s, the filter covariance matrix should he updated as if the measurement is available. This leads the measurements in the delay period to be fused as if z*(k) had been fused at time s. At time k, when z*(k) is available, incorporating z*(k) is then greatly simplified, by adding the following quantity after z(k)has been fused:

$$\delta\hat{x}_k = M_* k_s (z_k^* - H_k^* \hat{x}_s) \qquad (14)$$

If the delay is zero, M* is the identity matrix. For n>0, $M_*$ is given by:

$$M_* = \prod_{i=0}^{N-1} (I - k'_{k-i} H_{k-i}) F_{k-i-1} \qquad (15)$$

The prime on $K'$ signifies that these Kalman gain matrices have been calculated using a covariance matrix updated at time s with the covariance of the delayed measurement. As one factor in the above product can be calculated at each sample time, the method only requires two matrix multiplications at each sample time.

*Combination of Alexander and Adaptive Fading Method (our Suggestion)*

The mere principal of Alexander approach lies in the utilization of classic EKF. He initially assumes prior complete knowledge of the noise distribution characteristics, both for process and measurement signals. This proposition however, in many instances turns out to be nonrealistic.

To make room for the unknown variable noise attributes (mean and covariance matrices) and also for the realistic problem of model uncertainty, we provide an incremental strategy in which no such a restrictive prior assumption is made on noise properties, but rather we try to alleviate the degrading effects of unknown noise and model-reality mismatch, using an adaptive fading method fused with Alexander technique. Hence, based on above sections, the Equations (14) and (15) in Alexander method are replaced by

$$\delta\hat{x}_k = M_* \frac{\lambda_s}{\alpha_s} k_s (z_k^* - H_k^* \hat{x}_s) \qquad (16)$$

$$M_* = \prod_{i=0}^{N-1} \left(I - \frac{\lambda_{k-i}}{\alpha_{k-i}} k'_{k-i} H_{k-i}\right) F_{k-i-1} \qquad (17)$$

Where $\alpha_k$ and $\lambda_k$ are (10) and (12).

### Different Sampling Rates

Assume that the m sensors are geographically distributed on an industrial plant. Estimation procedure should have the ability to deal with the amount of data that will be received by the estimation node at different times. In this paper decentralized data fusion will be used to estimate states from asynchronous different sampling rate sensors.

In order to deal with different sampling rate sensors, Alouani suggestion (Alouani 1994) is to fuse received measurement values at the end of pre-selected time interval. In contrast, the authors suggestion is to fuse data of multi-rate sensors at any time in which a measurement is received. At this step, the fused value of states and covariances are sent back to estimation nodes as the information of data of time t=K and the procedure will be repeated to estimate the desired values of time t=K+1. As the sensors are multi-rate, naturally measured values of all sensors cannot be accessible at each arbitrary time-step. Thus, for those sensor nodes that measured value are not available, they send state and covariances which have been computed by only time-update equation to fusion node.

For i = 1, 2, . . . , m, suppose that $\widehat{X}_i(k)$ and $P_i(k)$ are the state estimates and the estimation error covariance matrices of $X_m(k)$ by adaptive fading extended Kalman filter, which are independent of each other, then the optimal fused estimate in the sense of linear minimum covariance is given by

$$\hat{X}(k|k) = \sum_{i=1}^{m} \alpha_{i,k}\, \hat{X}_{i,m}(k|k) \qquad (18)$$

Where

$$\alpha_{i,k} = \left(\sum_{j=1}^{m} P_{j,m}^{-1}(k|k)\right)^{-1} P_{i,m}^{-1}(k|k) \qquad (19)$$

and the corresponding estimation error covariance matrix is

$$P(k|k) = \left(\sum_{j=1}^{m} P_{j,m}^{-1}(k|k)\right)^{-1} \qquad (20)$$

## MATHEMATICAL MODEL OF CSTR

An irreversible and exothermic reaction A→B takes place inside the jacket CSTR. The reaction is operated by two proportional controllers that are used to regulate the outlet temperature and the tank level. A cooling jacket surrounds the reactor and the coolant is water in this case. Negligible heat losses, constant densities, perfect mixing inside the tank and uniform temperature in the jacket are assumed.

The dynamic equations describing the system are given by (Jones and Luyben 1989):

$$\frac{dV}{dt} = F_i - F_o \qquad (21)$$

$$\frac{d(VCa)}{dt} = F_i Ca_i - F_o Ca - V\left(k_0 \exp\left(\frac{E_a}{RT}\right)\right)Ca \qquad (22)$$

$$\rho c_p \frac{d(VT)}{dt} = \rho c_p (F_i T_i - F_o T) - \Delta H V\left(k_0 \exp\left(\frac{E_a}{RT}\right)\right)Ca - Ua_0(T - T_j) \qquad (23)$$

$$\rho_j V_j c_j \frac{dT_j}{dt} = \rho_j c_j F_j (T_c - T_j) + Ua_0(T - T_j) \qquad (24)$$

## SIMULATION STUDIES

For computer simulation, the CSTR nonlinear model is implemented using s-function and SIMIULINK facilities in MATLAB. The basic time unit is hours (hr) and the sampling time is taken to be equal to 0.005 hr.

As it is clear from CSTR equations, the outputs of the system are volume and temperature of product, concentration of A, and temperature of CSTR jacket. For the simulation studies, measurements (V,T) have been assumed as the observed values in order to estimate all states of the system (V, T, Ca, Tj ).

### Asynchronous Communication Delay

The implementation results of 3 proposed methods in corresponding section on CSTR case study are depicted in figures 1-2. In order to investigate the capability of the proposed methods in estimation of system states according to realistic settings in which asynchronous sensor data are corrupted with unknown noise together with an imperfect modeling, both incorrect values of noise variance assumptions and different communication delays are embedded in the simulation. The ratios of the incorrect values to the correct ones are 0.1 for measurement noise in Figure 1-2. These methods should compensate for the effect of lack of information. The capabilities of presented methods in extracting of real values are clearly illustrated via figures and root mean square error criteria (RMSE).

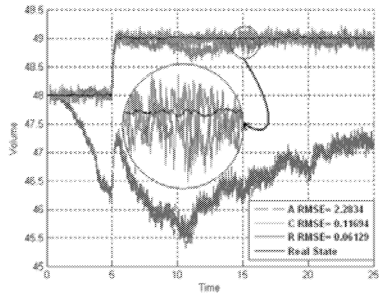


Figure 1. Estimation of Reactor volume which is measured by a 30 sampling period delay.

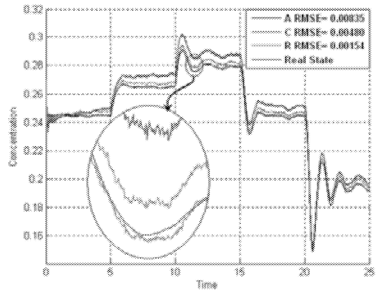


Figure 2. Estimation of Reactant concentration embedding a delay of 30 sampling periods.

Table 1. RMSE of different methods in estimation of CSTR volume when data of sensor V received after different values of delay .The measurement noise variance used in estimation is artificially set as 0.1 actual noise variance($(0.002\times48)^2$).

| Number of delay / RMSE | 2 | 5 | 10 | 15 | 20 | 30 | 50 | 80 | 100 |
|---|---|---|---|---|---|---|---|---|---|
| Alexander Method(A) | 2.0785 | 2.1003 | 2.10456 | 2.1101 | 2.1733 | 2.2001 | 2.26 | 2.3305 | 2.5412 |
| CAAF Method(C) | 0.3262 | 0.32548 | 0.32554 | 0.32551 | 0.32548 | 0.32512 | 0.32431 | 0.32357 | 0.32328 |
| Adaptive Recalculation Method(R) | 0.0595 | 0.06216 | 0.06639 | 0.07085 | 0.07525 | 0.08378 | 0.09783 | 0.11898 | 0.13223 |

Table 2.RMSE of estimation of temperature with different sampling rates.(NH corresponds to temperature sampling rate as high-rate sensor and NL is for volume as low rate sensor; NH/NL is the ratio of two quantities.)

| $\frac{NH}{NL}$ / RMSE | 2 | 5 | 8 | 10 | 15 | 20 | 30 | 40 | 50 | 80 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Low Rate(Volume) | 0.53242 | 0.53245 | 0.53247 | 0.53253 | 0.5326 | 0.53261 | 0.53273 | 0.5328 | 0.53294 | 0.53299 | 0.53301 |
| High Rate(Temperature) | 0.51294 | 0.51295 | 0.51296 | 0.51296 | 0.51299 | 0.51305 | 0.51323 | 0.51334 | 0.51345 | 0.51361 | 0.51382 |
| Fusion | 0.46142 | 0.46143 | 0.46145 | 0.46151 | 0.46153 | 0.46157 | 0.46166 | 0.46171 | 0.46179 | 0.46184 | 0.4619 |

Alexander method as discussed is only suitable in asynchronous paradigm with assumption of complete availability of noise information. This drawback has been approximately improved in CAAF method. On the other hand, Adaptive fading recalculation method has manifested the best performance but it is computationally more expensive than two other methods. In order to provide a more comprehensive comparison between the different methods in terms of accuracy, the RMSE values are presented for different values of delay from sensor V in tables 1.

### Sensors with Different Sampling Rate

Figures 3-4 show the results of proposed method in corresponding section in which data generation rate of sensor T (as a high rate sensor) is 30 times of sensor V(as a low rate sensor) .

Presented data in table 2 are produced by MATLAB software in which sampling rate of sensor T is NH/NL times of sensor V.

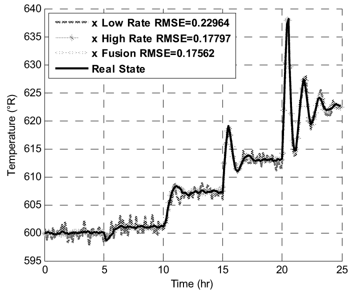


Figure 3. Estimation of Reactor temperature.

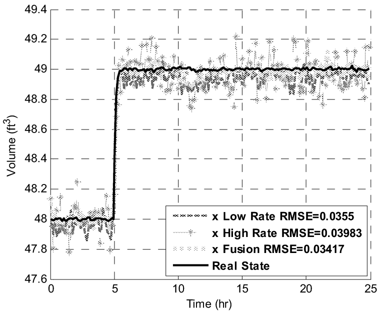


Figure 4. Estimation of Reactor volume.

## CONCLUSION

The mechanism of fusing uncertain, incomplete and asynchronous data from a variety of heterogeneous sensors to extract a single compilation of the overall system status for monitoring, control and decision making purposes has been considered in this paper. The inaccurate estimation of states in Alexander method due to a lack of solid consideration of existing model uncertainty and also a nonrealistic pre-assumption on noise distribution matrices, has been alleviated by incorporation of an adaptive fading EKF method (AFEKF). As compared with the adaptive recalculation method, the CAAF method proposed in this paper is much more computationally attractive, especially when the delays are large.

Also, an asynchronous decentralized data fusion platform for an estimation of multi-sampled signals based on AFEKF has been studied. The fused estimate proposed in this paper exhibits an improved performance over the scheme in which a single AFEKF is utilized for each sensor.

The simulation results, demonstrate the capability of the proposed approaches in real time monitoring and state estimation of a CSTR as a nonlinear case study.

## REFERENCES

R. L. Moose, M. K. Sistanizadeh, Gisli Skagfjord."Adaptive State Estimation for a System with Unknown Input and Measurement Bias."*IEEE Journal of Oceanic Emgineering*, vol. OE-12, NO. I,Jan. 1987

R. L. Moose,"An adaptive state estimation solution to the maneuvering target problem."*IEEE Trans, Automat. Contr.,* vol. AC-20, pp.359-362, June 1975.

K. H. Kim, J. G. LEE, C. G. Park ,"Adaptive Two-Stage Extended Kalman Filter for a Fault-Tolerant INS-GPS Loosely Coupled System."*IEEE Transac on Aerosapce and Electronic Systems*, vol. 45, NO.1 January 2009.

H. L. Alexander, "State estimation for distributed systems with sensing delay". *In SPIE* vol. 1470 Data Structures and Target Classijication, 1991.

R. J. Fitzgerald, "Divergence of the Kalman filter", *IEEE Trans. Automatic Control*, Vol. AC-16. No. 6, pp. 736- 747, 1971.

K. H. Kim, *An Adaptive design for a fault tolerant navigation system*. Ph.D. dissertation, Seoul National University, Seoul, Korea, 2006.

T. D. Larsen, N. A. Andersen and O. Ravn,"Incorporation of Time Delayed Measurements in a Discrete-time Kalman Filter", *IEEE*, pp.3972-3977,1998.

A. T. Alouani, "On Asynchronous Data Fusion.", *IEEE*,1994.

Jones, W. L. Luyben, *Process Modeling Simulation and Control for Chemical engineers*, McGraw-Hill,2nd edition,1989 .

# DECISION SUPPORT SYSTEMS

# DESIGN OF A DECISION-MAKING SUPPORT WITHIN AGENT-BASED SIMULATIONS REFLECTING RAILWAY TRAFFIC

Antonín Kavička and Michael Bažant
Department of Software Technologies
Faculty of Electrical Engineering and Informatics
University of Pardubice
Studentská 95, CZ-532 10 Pardubice
Czech Republic
E-mail: Antonin.Kavicka@upce.cz, Michael.Bazant@upce.cz

## KEYWORDS

Agent-based simulation, decision-making support, artificial neural network, passenger railway station.

## ABSTRACT

The paper deals with the problem of decision-making supports, which are supposed to be (i) properly integrated into the simulation models and (ii) potentially prearranged (verified) before their employment in simulation experiments. It was selected the architecture called ABAsim (Agent-Based Architecture of simulation models) to demonstrate a suitable embodiment of advisory components (supporting decision-making processes) within an agent. The case study briefly presents traffic simulations related to a passenger railway station, within the frame of which a selected decision-making support (realised as an artificial neural network) is introduced.

## INTRODUCTION

Implementation of decision-making supports and their proper integration into the simulation models represents quite challenging problem. The solution of that problem influences the credibility of the entire model. At first, let us present the mentioned topic using a specific agent-based simulation architecture (called *ABAsim*). Next, the simulation model (exploiting ABAsim architecture) reflecting the traffic within the passenger railway station is briefly introduced. Finally, a typical decision-making problem (of platform track assignment) occurring within passenger railway stations is presented and the corresponding realised support (including its integration into the model) is described.

## ABASIM ARCHITECTURE IN A NUTSHELL

Agent-based architecture *ABAsim* (described in detail in (Kavička et al. 2007)) was mainly developed for simulation of large service systems (namely transportation logistic systems). Let us mention the most important features of those systems, which essentially influence the architectural properties.

Service system structures can be considered (from the point of view of order execution) strictly hierarchical. The order (the customer) entering the system initiates a recursive sequence of suborders, according to the rules of competence redistribution. All system elements (subsystems) work in a synergic way (unlike the majority of natural systems).

The entities within a service system (orders/customers and resources) can be divided into specialised classes with the same behavioural rules for all entities in the class. This means that the responsibility for the behaviour of these entities is taken over by their superior subjects (agents). Hence, there is no reason to consider individual entities as agents. Transport service systems usually represent large-scale systems. It is necessary in most cases to transfer service resources to the customer (or vice versa), in order to actualise the service activity. Hence, frequent and complex transposition processes are typical within such systems.

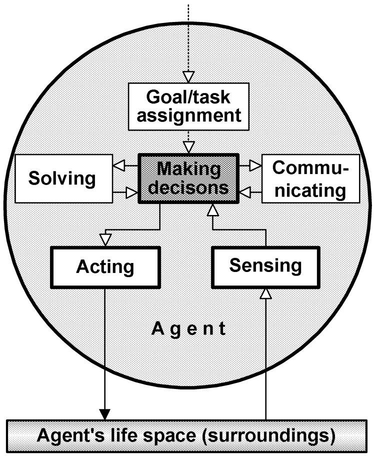


Figure 1: The agent functions

The main agent functions in ABAsim architecture are described in Figure 1. A given task or goal is assigned to each agent. The agent then realises, according to its mission, its own life-cycle: *sensing – decision making – acting* (within its life space) using the support of solving (focused on making solution proposals) and *communicating* with other agents (eventually with human operators). If the agent detects a problem or a situation beyond its delegated competence, it informs other agents about the need for a corresponding solution.

Each agent can be decomposed into the following groups of internal components (Figure 2). The first, control and decision making component (called the *manager*) is responsible for making decisions and for inter-agent communication. In addition, the manager represents the central agent component because it initiates the work of other internal components and can also communicate with all of them.

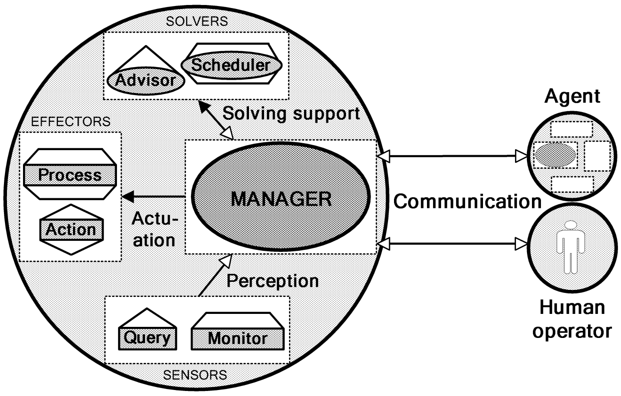


Figure 2: The internal structure of an agent

The group of *sensors* is specialised for mining information from a state space. This group is composed of two kinds of components - the *query* delivers the required forms of information instantly, and the *monitor* scans the state space in some time interval and continuously brings important information to the manager.

The next group, called *solvers*, provides solutions of problems to the manager, which can accept them or ask for alternative ones. The *advisor* is a passive component able, by return, to react only on the manager's requests for delivery of proposals for problem-solving. On the other hand, the *scheduler* (focused on a restricted scope of problems) works continuously for the manager, on the basis of either a priori rosters or schedules. Thus, the solvers represent the components, within the frame of which the decision supports are involved.

The last component group includes *effectors* (actuators), which make changes to system status after receiving corresponding instructions from the manager. No other agent components are allowed to make these changes. An *action*-component makes prompt state changes (within one instant of simulation time), while a *process*-component makes them continuously (within an interval of simulation time) until its task is finished.

Simulation models of simple real systems could be composed of only one agent; however, the simulation of complex service systems requires obviously a multi-agent approach, using the agents within some organisational structure. Let us remark that the philosophy of ABAsim architecture was also partly inspired by the paradigm of reactive agents (Brooks 1991), which is based on a society of reactive rather than proactive agents. The intelligence of such society emerges when one observes the whole community and not its separate members (individually of relatively low intelligence).

To summarise the philosophy of ABAsim operation: The control role is played by mutually communicating managers (supported by sensors and solvers), which initiate the activities of effectors at the correct time instants and under particular conditions.

## TRAFFIC SIMULATIONS OF A PASSENGER RAILWAY STATION

Significant progress has been achieved during the last decade in the area of simulations reflecting the traffic within railway stations (Adamko and Klima 2008), (Kavička et al. 2006), (Kavička and Bažant 2007), (Bendfeldt et al. 2000), (HaCon 2007), (Nash and Hürlimann 2004). Current research and development pays special attention to a support of decision making procedures related to operational problems occurring within (simulation models of) railway stations (Chakroborty 2008), (Bažant 2008), (Bažant and Kavička 2009). The realisation of relevant decision supporting components can be based on different approaches, methods and paradigms.

The system of a passenger station can be classified (from the viewpoint of investigation concentrated on railway traffic) as a queuing system. Resources are represented as tracks, service personnel members and locomotives. Passenger trains (or train sets) participating in the station traffic and being objects of an attendance are understood as customers.

Let us introduce the hierarchical structure of agents (Figure 3) constituting the simulation model of a passenger railway station. *Dispatcher* represents a topmost agent responsible for all agents, which is directly superior to *Surroundings* agent (reflecting relevant processes within surroundings of examined system, e.g. train arrivals), *Service* agent (responsible for the execution of user-defined technological procedures) and *User Cooperation* agent, which stands for humans interacting with the model during run-time. In addition, *Dispatcher* carries out its own activities (e.g. associated with the arrivals and departures of trains). Common *Resource* agent governs a set of agents (*Infrastructure*, *Personnel* and *Engines*) responsible for allocation, work and release of respective resource types. *Operation* agent organises (delegating specialised tasks to its subordinated agents) executions of elementary technological activities. *Transfer* agent realises individual movements of train sets using a corresponding rail infrastructure. And finally, the agent of *Regroup* operations controls the reorganisations related to compositions of train.

Simulation model of a passenger railway station include quite a wide range of decision-making tasks. Hereafter attention is paid only to one selected problem related to assignment of platform tracks to delayed trains. The mentioned problem lies within the authority of *Dispatcher* agent.

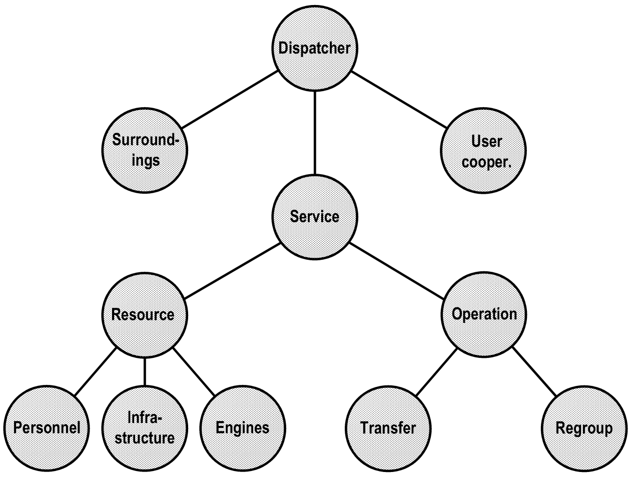


Figure 3: The hierarchical structure of agents

## TRACK ASSIGNMENTS TO DELAYED TRAINS

Assignment of platform track to an arriving train represents a typical decision making task for dispatchers within passenger railway stations. If the inbound trains follow the timetable, the platform tracks are commonly assigned according to a priori created plan. In the case of a delayed arriving train the dispatcher is supposed to make an operative decision (potentially considering a set of substitutive tracks) about a relevant platform track assignment.

The above mentioned assignment problem should be properly solved also within particular simulation models. This is important for example, in the case of investigations focused on passenger stations suffering from frequent delays of arriving trains. Assigned tracks ought to correspond to resulting decisions made by experienced station dispatchers in reality. After assigning a platform track to a relevant train many other specialised algorithms (involved within a simulation model) are carried out (e.g. an algorithm focused on a setting train route respecting the rules of an interlocking system, an algorithm calculating the dynamics of the train movement to the assigned platform etc.).

A station dispatcher (managing real railway traffic) partially subjectively evaluates potential platform tracks that are suitable for an assignment to a delayed incoming train. The ultimately assigned track represents the best solution according to the expert knowledge of a particular dispatcher using certain criteria. The same strategy is applicable for a relevant simulation model. The first stage of the original submitted approach is focused on delayed trains from one arrival direction only. Platform track selection is primarily related to construction of an *a priori track set*, the elements of which can be admissibly assigned to a considered (delayed) inbound train. The tracks contained in the mentioned set are determined with regard to the defined arrival and departure line track. The sets can be further reduced according to the specific conditions (e.g. some elements/tracks are removed from the relevant set because of their insufficient length with respect to the considered train etc.).

The next step is associated with the final selection of a particular platform track (from an a priori set), which represents the most suitable solution (according to specific criteria) for the considered train in time of its real arrival. The mentioned criteria take into account the knowledge of station dispatchers and are formed as follows:

*A:* Track vacancy degree at the moment of train arrival.
*B:* Track vacancy period with regard to station sojourn time of an arriving train.
*C:* Occupation of the neighbouring track at the same platform (owing to the selected track) by a connection train.
*D:* Further technical and technological preferences of the track in respect to an arriving train.

A specified track assignment problem is obviously connected with multiple-criteria decision making focused on selection of variants (Figueira et al. 2004). The finite set of variants corresponds to the above mentioned a priori track set containing the tracks, which stand as candidates for a relevant assignment. If the criteria are at our disposal (*A*, *B*, *C*, *D*) and it is possible to calculate the criterion values (the relevant calculation is described in (Bažant and Kavička 2009)) of investigated decision variants then a criterion matrix can be created. An element of criterion matrix $y_{ij}$ expresses the value of a criterion $i$ (where $i = 1, 2, ..., 4$ reflects criteria *A*, …, *D*) for the relevant variant/track $k_j$. The mentioned matrix can be formalised as follows.

$$\begin{array}{cc} & \begin{matrix} k_1 & k_2 & \dots & k_m \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \end{matrix} & \begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1m} \\ y_{21} & y_{22} & \cdots & y_{2m} \\ y_{31} & y_{32} & \cdots & y_{3m} \\ y_{41} & y_{42} & \cdots & y_{4m} \end{pmatrix} \end{array} \qquad (1)$$

When evaluating values of individual criteria, the *maximisation principle* is applied, i.e. the criteria are designed so that a particular variant is the best when the highest relevant criterion values are used. Calculation of the criterion values related to criteria *A* and *B* utilises a static *track occupation plan*, which is constructed for every major passenger railway station. Each running track within the station is linked with data (using the time step of one minute) about its occupation of trains.

Attention is further paid to an artificial neural network applying supervised learning (Nguyen et al. 2003), which represents one of the possible ways of solving the discussed track assignment problem. The mentioned neural network requires a prearrangement of two sets: a set of specific individual traffic situations (training patterns/inputs) and another set of relevant expert solutions. Produced outputs of a trained neural network are then compared with corresponding expected solutions – an expert/supervisor continuously evaluates the quality of the outputs and decides upon the next training steps.

Selection of an appropriate neural network type (e.g. feed-forward network, multilayered perceptron etc.) represents an essential problem. It is quite difficult to determine the suitable kind of neural network (concerning a given problem) in advance. Thus, experiments with different kinds of neural networks and their diverse parameterisations were carried out. As a result of the experiments it was claimed that a two-layered perceptron produced the most encouraging outcomes.

The above mentioned methodological approach can be divided into the following steps:

- Gaining knowledge about the platform track assignment problem.
- Specification of the calculation method applied to getting criteria (*A–D*) values.
- Computation of criterion matrices (exploiting criteria *A–D*) for different traffic situations.
- Separation of available data into disjoint sets (training set and test set).
- Supervised learning of selected neural network using data from the training set.
- Evaluation of the neural network behaviour in respect to input data from the test set.

A two-layered perceptron was trained, tested and then applied to platform track assignments for arriving trains within the simulation model reflecting the system of a passenger station.

## CASE STUDY

Selection of Prague main station as a testing case took into account (i) the number of platform tracks, (ii) the number of trains approaching the station within the peak hours and (iii) good knowledge of local operational conditions (especially related to control and decision-making processes). Attention was paid to the timetable 2004/2005 and the contemporary track infrastructure layout. Traffic at peak time (5.30–9.00 a.m.) was investigated in order to verify correct behaviour of the neural network. Delayed trains arriving within the frame of that rush period cause the most serious problems (comparing with the rest of the day) connected with platform track assignment.
One arrival direction was chosen to inspect neural network sufficiency of a correct track assignment. Local station dispatchers recommended taking the arrival direction from the town Kolín, which disposed of the highest number of delayed trains (because of reconstruction works within relevant line) from all arrival directions concerning the studied station. Eleven long distance trains arrive at Prague main station (using one arrival track from the mentioned direction) during morning rush hours, thus corresponding criterion matrices (1) for all these trains were evaluated. The main concept of investigation was to examine a sufficient number and lengths of delays related to each train. For that reason the delay values of each train varied from 0 to 60 minutes (with the step of one minute).

Different delay values of arriving trains are connected with diverse traffic situations (different trains occupy station tracks) depending on time. In this connection, 671 criterion matrices were elaborated considering 61 delay values for each of the 11 trains. All other trains, observed within simulated traffic, were going on time according to the mentioned timetable. Next, an expert determined a platform track (expected solution), which would be assigned in reality, for each calculated matrix (pattern) reflecting a traffic situation.

The linearly ordered initial set of applied data (reflecting traffic situations associated with 671 instances of delays related to eleven selected trains) was divided into the two following disjoint subsets:

- *Training subset* (containing 336 elements/patterns) constructed on the basis of systematic sampling applied to the initial set (sampling period was set to 2 minutes).
- *Test subset* (composed of 335 elements) was constructed analogically; particular sampling was shifted in comparison with the previous subset (phase-shift was equal to 1 minute).

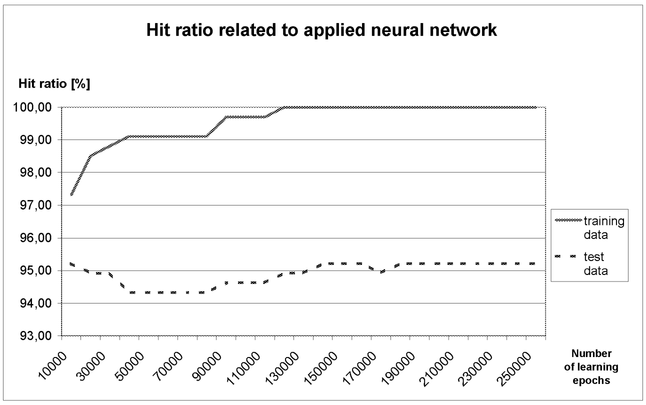


Figure 4: Hit ratio trends of neural network behaviour

The process of supervised learning coupled with a two-layered perceptron, utilised data elements from the training subset. The neural network was able to learn all patterns after applying a reasonable number of learning epochs (Figure 4). Next, the network was tested using data inputs from the test set. The testing stage reached a 95 % hit ratio (comparing the network outputs and expert expectations). Different track assignments from an expert's proposal were obtained in five percents of cases (concerning test data). In spite of this fact, even those “unsuccessful” outputs (platform track assignments) still represented quite good solutions.

## CONCLUSIONS

The discussed decision-making support was realised as the *advisor-component* contained in the *Dispatcher* agent. That component is activated for each (potentially delayed) incoming train. The relevant criterion matrix (reflecting the current operational situation) is calculated and utilised as an input for the trained artificial neural network. The network produces the outputs evaluating the fitness rates (with regard to an assignment to the given train) of all considered platform tracks. Finally, *Dispatcher* agent selects the track with the highest fitness rate and assigns it to the relevant arriving train. The mentioned approach to integration of the decision supports into agent-based simulation models seems to be very efficient because it is quite easy to identify a competent agent, within the frame of which the particular support is supposed to be located. In addition, the approach contributes to a flexible construction and maintenance of simulation models.

Utilisation of a two-layered perceptron as a support of track assignments to delayed arriving trains (within the simulation study investigating Prague main stations) can be classified as successful. Alternatively realised supports, based on classical mathematical methods related to multiple-criteria evaluation (focused on selection of variants) reached the results comparable to the neural network outputs (Table 1) however the application of mathematical methods is associated with the problematic subjective expert quantification of relations between individual criteria (the pairwise comparison matrix (Figueira 2004) is typically utilised). The mentioned quantification usually complies with the limited number of considered operational situations, the solutions of which are very good. However, the solutions of other situations can be potentially unacceptable, i.e. in this case the new quantifications have to be realised. On the contrary, the artificial neural network has a higher potential of generalisation – especially in the case that the wide range of training patterns (traffic situations) are available.

Table 1: Hit ratio evaluated by classical mathematical methods (Bažant 2008)

| *Method\criteria weight* | *Criteria weight* | | | | *Hit ratio [%]* |
|---|---|---|---|---|---|
| | *A* | *B* | *C* | *D* | |
| Sequence method | 0,40 | 0,30 | 0,20 | 0,10 | 92,25 |
| Fuller method | 0,35 | 0,35 | 0,20 | 0,10 | 93,14 |
| Geometric mean method | 0,44 | 0,44 | 0,09 | 0,03 | 99,55 |
| Saaty method | 0,44 | 0,44 | 0,10 | 0,03 | 99,25 |

In spite of the fact that the developed approach currently takes into account the delayed trains from one arrival direction only, it already represents a substantial benefit for selected types of simulation studies. For example, an investigation of passenger stations suffering from frequently delayed trains mainly arriving from one direction can be mentioned (e.g. because of long-term reconstruction works within relevant railway line). On the other hand, there are only a few delayed trains coming from other arrival directions, i.e. the delays of those trains are not considered (certainly after relevant expert review) within this particular simulation model. Thus, the discussed approach is effective for solving the mentioned kinds of problems.

The next research stage is supposed to upgrade the standing approach, i.e. the delayed trains incoming from any arrival direction can occur.

## ACKNOWLEDGEMENT

This work has been supported by the National Research Program of the Czech Republic under project "MSM 0021627505 Theory of Transportation Systems".

## REFERENCES

Adamko, N., Klima, V. 2008. "Optimisation of Railway Terminal Design and Operations using Villon Generic Simulation Model". *Transport*, No.4, 335-340.

Bažant, M. 2008. "Platform track assignment for delayed train using mathematical methods related to multiple-criteria evaluation". In *Proceedings of the 16th International Symposium ŽEL 2008* (Žilina, Slovak Rep., Jun. 4-5). University of Žilina, 325–332.

Bažant, M., Kavička, A. 2009. "Artificial neural network as a support of platform track assignment within simulation models reflecting passenger railway stations". *Journal of Rail and Rapid Transit* (in Press).

Bendfeldt, J-P., Mohr, U., Müller, L. 2000. "RailSys, a system to plan future railway needs", *Computers in Railways VII.* WIT Press, Southampton (UK), 249-255.

Brooks, R. 1991. "Intelligence without representation". *Artificial Intelligence*, No. 1-3 (vol 47.), 139-159.

Chakroborty, P., Vikram, D. 2008. "Optimum assignment of trains to platforms under partial schedule compliance". *Transportation Research Part B: Methodological*, No. 2, 169-184.

Figueira, J., Greco, S., and Ehrgott, M. 2004. "Multiple Criteria Decision Analysis: State of the Art Surveys". Springer Verlag, New York (US).

HaCon. 2007. "RASIM - Simulation system for the analysis and planning of railway systems". *http://www.hacon.de/rasim/index.shtml.*

Kavička, A., Bažant, M. 2007. "Simulations as a support for planning infrastructure within Prague Masaryk station". In *Proceedings of 21st European Conference on Modelling and Simulation* (Prague, Czech Rep., Jun. 4-6). European Council for Modelling and Simulation (UK), 363-367.

Kavička, A., Klima, V., Adamko, N. 2006. "Analysis and optimisation of railway nodes using simulation techniques". *Computers in Railways X.* WIT Press, Southampton (UK), 663-672.

Kavička, A., Klima,V., Adamko, N. 2007. "Simulations of transportation logistic systems utilizing agent-based architecture". *International Journal of Simulation Modelling*, No.1, 13-24.

Nash, A., Hürlimann, D. 2004. "Railroad Simulation using OpenTrack", *Computers in Railways IX.* WIT Press, Southampton (UK), 45-54.

Nguyen, H. et al. 2003. "A first course in fuzzy and neural control". CRC Press, Boca Raton, Fl.

# DEVELOPMENT OF DECISION SUPPORT AND SIMULATION SYSTEM BPsim.DSS: INTEGRATION OF SIMULATION, EXPERT, SITUATIONAL AND MULTI-AGENT MODELING

Konstantin A. Aksyonov
Eugene A. Bykov
Olga P. Aksyonova
Wang Kai
Alexey V. Popov
Ural State Technical University
Mira street, 32
Ekaterinburg, 620000, Russia
E-mail: wiper99@mail.ru

Elena F. Smoliy
Ekaterina M. Sufrygina
Irina A. Spitsina
Alexey A. Sheklein
SPC «Business support systems», Ltd.
Mira street, 32
Ekaterinburg, 620000, Russia
E-mail: mail@bpsim.ru

**KEYWORDS**
Decision support, simulation, expert system

## ABSTRACT

Paper focuses on integration of simulation modeling system and decision support system in order to achieve efficient processes design in organizational-technical systems. Paper describes theoretical aspects of system development, used mathematical model and methods, stops in detail on knowledge representation model, describes multi-agent resource conversion processes approach, combined with InteRRaP architecture, continues with development of software, implementing the designed mathematical model, and concludes with developed software application and economical effect.

## INTRODUCTION

Use of situational models in process control facilitates efficiency and taken decisions quality growth, decision taking time decrease, resource consumption rationalization. Dynamic situations modeling systems (DSMS) design is one of perspective directions of decision support systems (DSS) development. Currently multi-agent systems area is under major research; one of its features is agent collaborations interaction, such agents identify decision making people. An important area of multi-agent technologies application is simulation. Multi-agent systems engineering approaches can be distinguished into two types:

1. Based on object-oriented methods and technologies and
2. Use of traditional knowledge engineering methods (Andreichikov and Andreichikova 2004).

Object-oriented method extensions and multi-agent systems engineering technologies are developed in methodologies of first type. Certain CASE tools support information systems development based on object-oriented methods (All Fusion, Rational Rose). Object oriented agent behavior definition language UML-RT is utilized in multi-agent simulation system AnyLogic. Methodologies of second type are built on basis of traditional knowledge engineering methods extension. An actual task is development of dynamic situations modeling system, based on object-oriented technologies.

## CURRENT STATE OF DYNAMIC SITUATIONS MODELING SYSTEMS

Dynamic situations modeling systems area state analysis reveals unavailability of resource conversion processes oriented systems. Nearest functionality analogs include simulation and expert modeling tools, particularly real-time expert system G2, multi-agent simulation system AnyLogic, business-processes modeling system ARIS, simulation system Arena, dynamic situations modeling system BPsim.MAS and technical economical engineering system BPsim.MSS. We have cade a comparison of these products. The results revealed, that the named systems lack support of some features, useful in effective simulation. For example, agent-based approach implementation is limited. Another disadvantage of two most powerful systems, ARIS ToolSet and G2, is a very high retail price, which might stop a potential customer.

Simulation, situational modeling and expert systems are used in modeling, analysis and synthesis of organizational-technical systems and business processes. Multi-agent resource conversion processes theory (Aksyonov and Goncharova 2006) may be used for organizational-technical systems definition from decision support point of view, a dynamic component of business processes, expert systems, situational modeling and multi-agent systems. Decision support system development requires selection or development of mathematical apparatus.

## KNOWLEDGE REPRESENTATION MODEL

One of the most important problems in intelligent systems engineering is selection and design of subject area knowledge representation models, that allow the easiest native transition from nonformalized knowledge and views to formal models and knowledge base. Knowledge elicitation and acquisition process might be very complicated in intelligent systems engineering.

Knowledge structuralization complexity is revealed in requirement for subject area model, that allows the most adequate transition to technical implementation with least effort (Shvetsov 2004).

Subject area conceptual model needs subject area structure to be defined, available objects and subjects behavior determined, logical interaction models designed. Minsky defined a frame as a structure for stereotyped (standard) situations representation (Minsky 1975). This structure is filled with various information: defining objects and events expected in certain situations as well as providing guidelines on use of information, contained in a frame. Main idea is to concentrate all knowledge, related to specific objects or events class, in a common data structure, but not to distribute it between a multitude of small structures like logical formulae and productive rules. Such knowledge is either concentrated within the structure itself or available from the structure (e.g. stored within a related structure) (Jackson 1998).

Each frame is associated with various information (including procedures), e.g. information defining frame use, expected results of frame execution, directions for actions when expectations are not fulfilled, etc.

Frame-based approach reveals the following advantages: frame concept is natively integrated with subject area conceptual modeling; frame structures are easily defined within object-oriented design; inheritance capabilities are effectively supported; subject area hierarchical representation is provided. Thus, frame-based approach selection might be a reason for object-oriented approach and object programming languages application in dynamic situations modeling system development. Such appoach minimizes costs for software development as well.

Analysis of Shvetsov's object-oriented design and programming-related research reveals three main model classes, implementing class-based representation formalism: semantic networks and frames-based models; database theory and semantic data patterns-based models; abstract data types-based models. Frame-based languages extend semantic and object-oriented data models capabilities, which substantiates their application and further research in this area.

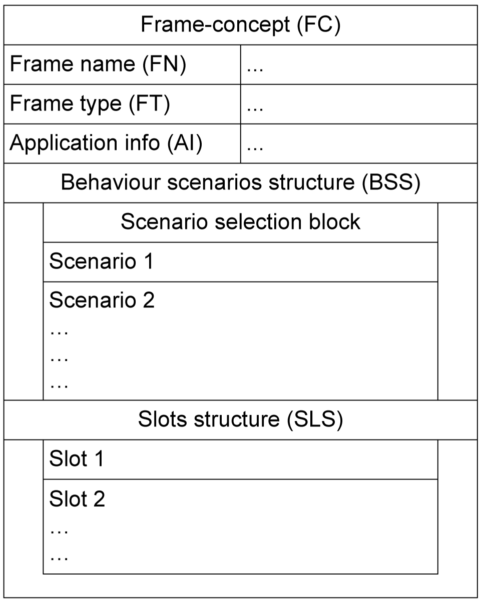

Frame-concept (FC)
Frame name (FN) ...
Frame type (FT) ...
Application info (AI) ...
Behaviour scenarios structure (BSS)
Scenario selection block
Scenario 1
Scenario 2
…
…
…
Slots structure (SLS)
Slot 1
Slot 2
…
…

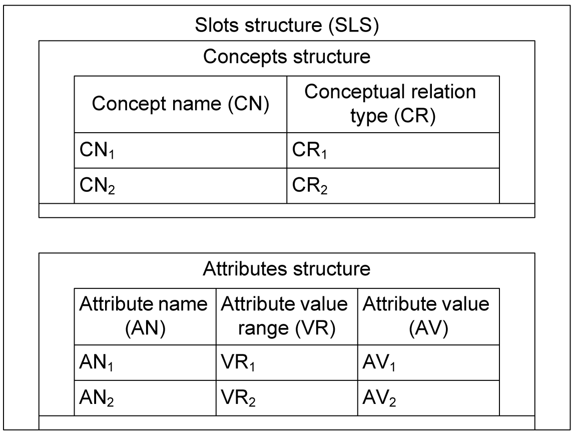
Figure 1: Frame-concept construction

This research makes use of frame approach, based on frame-like structures association with J.F.Sowa's conceptual graphs constructions (Sowa 2000), in order to design subject area conceptual model and achieve software development costs reduction. Active and passive frames distinction and agent behavior consideration are among approach advanteges.

Basic frame-concept (FC) construction is presented on Figure 1. Frame name is a unique identifier, used within subject area conceptual model.

Frame-concept level application information contains informal verbal definition of available frame-concept application situations, behavior scenarios, selection features, etc. Dynamic subject area components and agents behaviour is defined in behaviour scenarios structure, containing scenario selection block, that allows current frame alternative behaviour options generation.

Slots structure consists of two structures: concepts structure and attributes structure (Figure 2). Concepts structure contains list of frame-concepts, in some way embedded into or descendent from current frame-concept; relation type is indicated in «conceptual relation» field, being relation of specific concept (SC) $SC_i$ to current frame-concept FC, $SC_i$ is i-*th* frame-concept name. Frame-concepts are combined into conceptual graphs structures to form subject area logical organization.


Slots structure (SLS)
Concepts structure

| Concept name (CN) | Conceptual relation type (CR) |
|---|---|
| $CN_1$ | $CR_1$ |
| $CN_2$ | $CR_2$ |

Attributes structure

| Attribute name (AN) | Attribute value range (VR) | Attribute value (AV) |
|---|---|---|
| $AN_1$ | $VR_1$ | $AV_1$ |
| $AN_2$ | $VR_2$ | $AV_2$ |

Figure 2: Slots structure

Conceptual graph is a bipartite graph with two types of nodes: concept nodes or conceptual nodes and conceptual relations nodes. Thus, frame-semantic knowledge representation is used.

Frame-concept model is defined in the following way:

$$FC = \langle FN, FT, AI, BSS, SLS \rangle$$

$$SLS = \langle CS, AS \rangle$$

$$CS = \{(CN_1, CR_1), (CN_2, CR_2), \ldots, (CN_n, CR_n)\}$$

$$AS = \{(AN_1, VR_1, AV_1), (AN_2, VR_2, AV_2), \ldots, (AN_m, VR_m, AV_m)\}$$

– where *FC* – frame name, *FT* – frame type, *AI* – application information, *BSS* – behaviour scenario structure, *SLS* – slots structure, *CS* – concepts structure, *AS* – attributes structure, $CN_n$ – concept name, $CR_n$ – conceptual relation, $AN_m$ – attribute name, $VR_m$ – attribute available values range, $AV_m$ – attribute value.

Thus, frame-concept and conceptual graph-based approach to subject area definition allows frame-semantic knowledge representation model use.

Multi-agent resource conversion processes architecture design is described in next sections.

## MULTI-AGENT RESOURCE CONVERSION PROCESS MODEL

In this research, we will define the resource conversion process (RCP) as the process of an input conversion (resources necessary for process execution) into output (products – outcomes of process execution). The main objects of discrete Multi-agent RCP are the following: operations (*Op*), resources (*Res*), control commands (U*)*, conversion devices (*Mech*), processes (*PR*), sources (*Sender*) and resource receivers (*Receiver*), junctions (*Junction*), parameters (*P*), agents (*Agent*). Process parameters are set by the object characteristics function. Relations between resources and conversion device are set by link object (Relation). The agents existence resumes availability of the situations (Situation) and decisions (action plan) (Decision). Agents control the RCP objects. More detailed information about multi-agent resource conversion processes apparatus is presented in (Aksyonov et al. 2008).

Utilized simulation algorithm consists of the following stages: current moment of time identification $SysTime = \min_{j \in RULE} T_j$ ; agents actions processing; conversion rules queue generation and operation memory state modification. Simulator makes use of expert system unit for situations diagnosis and control commands generation (Aksyonov et al. 2008).

System graphs of large scale integration level (Avramchuk et al. 1988) are used for multi-agent resource conversion process hierarchical structure definition (Aksyonov and Goncharova 2006). Right selection and implementation of multi-agent architecture is a key factor in multi-agent object oriented decision support system design.

## MULTI-AGENT RCP EXTENSION WITH INTERRAP ARCHITECTURE

Two main agent architecture classes are distinguished. They are:

1. **Deliberative agent** architecture, based on artificial intelligence principles and methods, i.e. knowledge-based systems;
2. **Reactive** architecture, based on system reaction to external environment events.

All currently existing architectures cannot be defined as purely behavioral or purely knowledge-based. Any designed architecture is hybrid, offering features of both types.

Multi-agent resource conversion process architecture is based on InteRRaP (Muller and Pischel 1993, Aksyonov et al. 2009) architecture, as the most appropriate for subject area. InteRRaP architecture represents an aggregate of vertically ordered levels, relating to common management structure and using common knowledge base. Architecture consists of blocks: external environment interface, reactive sub-system, planning sub-system, cooperation with other agents, and hierarchical knowledge base. External environment interface defines agent capabilities in external environment objects and events perception, influenceability, and means of communication. Reactive sub-system utilizes agent capabilities in reactive behavior, as well as partly utilizes agent knowledge of procedural kind. It is based on «behavior fragment» concept as reaction draft in some standard situation. Planning sub-system contains planning mechanism that provides agent local plans capability (not related to co-operative behavior). Cooperation sub-system is responsible for building co-operative behavior plans, focusing on certain joint goals, or fulfillment of obligations for other agents, as well as implementation of agreements.

In accordance with InteRRaP architecture common concept, multi-agent RCP agent model is represented in four levels:

1. **External environment** model corresponds to the following MRCP elements: convertors, resources, tools, parameters, goals. External environment performs the following actions: generates tasks, transfers messages between agents, processes agent commands (performs resource conversion), alters current state of external environment (transfers situation $S_n$ into state $S_{n+1}$).
2. **External environment interface** and reactive behavior components are implemented in form of agent productional rules base and inference machine (simulation algorithm).
3. **Reactive behavior** components performs the following actions: receives tasks from external environment, places tasks in goal stack, collates goal stack in accordance with adopted goal ranging strategy, selects top goal from stack, searches knowledge base. If appropriate rule is located, component transfers control to corresponding resource convertor from external environment. Otherwise, component queries local planning sub-system.
4. **Local planning** level purpose is effective search for solutions in complex situations (e.g. when goal achievement requires several steps or several ways for goal achievement are available). Local planning component is based on frame expert system. Frame-concept and conceptual-graph based approach is utilized for knowledge formalization.

Subject area conceptual model and agent local planning knowledge base design is based on UML class diagram extension. Semantically this notion may be interpreted as definition of full decision search graph, containing all available goal achievement ways (pre-defined by experts). Current knowledge base inference machine is implemented in decision search diagram, based on UML sequence diagram. Each decision represents agent activity plan. Each plan consists of a set of rules from reactive component knowledge base. Based on located decision, current agent plan is updated. Examination of all available options, contained in knowledge base, generates agent plans library.

If an agent, when processing task or message received from external environment, is unable to locate appropriate rule in its knowledge base (e.g., select an option from several ones), the reactive behavior component queries plans library, indicating goal (i.e. task to execute, or external environment state to bring into). Planning sub-system searches plans library, selects appropriate plan and places first rule of selected plan into reactive component goals stack.

## INTELLIGENT AGENT OPERATION ALGORITHM

Special-purpose object-oriented language RADL (Reticular Agent Definition Language) in form of **When-If-Then** construction implemented in agents and multi-agent systems engineering system *Agent Builder* (Reticular Systems, Inc.) (Andreichikov and Andreichikova 2004) is used as a basis for agent behaviour rules. Mental model includes intentions, desires, obligations and capabilities as well as agent behaviour rules definition. Specific intelligent actions are calculated on the basis of this model.

## DECISION SUPPORT SYSTEM BPSIM.DSS

Object-oriented decision support system BPsim.DSS is implemented on basis of dynamic situations modeling system BPsim.MAS and technical economical engineering system BPsim.MSS integration. Agent model is represented with four levels in compliance with InteRRaP architecture general concept. *External interface* and *reactive behavior* components together with *external environment* model are implemented in BPsim.MAS tool. *Local planning* component is based on BPsim.MSS expert system module. Expert system shell visual output mechanism builder is based on decision search diagrams (UML sequence diagram extension) and presented on Figure 3. Cooperation level is based on both modules.

Object-oriented decision support system BPsim.DSS allows the following features implementation: subject area conceptual model definition, multi-agent resource conversion process dynamic model design, dynamic simulation, experiment results analysis, reporting on models and experiment results, data export to MS Excel and MS Project.

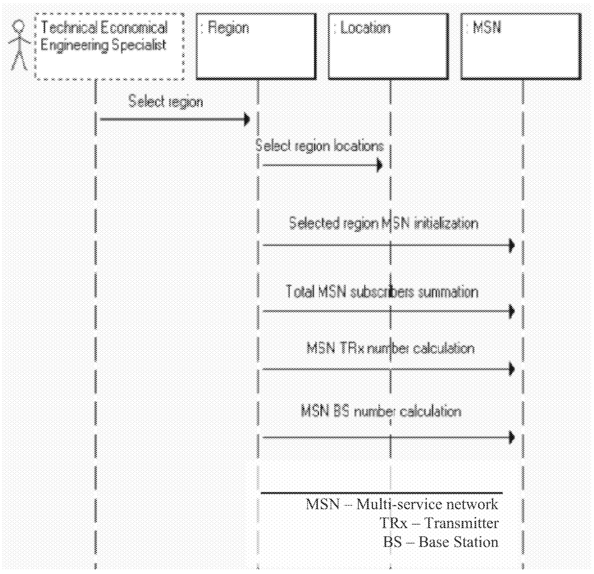

Technical Economical Engineering Specialist
: Region
: Location
: MSN
Select region
Select region locations
Selected region MSN initialization
Total MSN subscribers summation
MSN TRx number calculation
MSN BS number calculation
MSN – Multi-service network
TRx – Transmitter
BS – Base Station

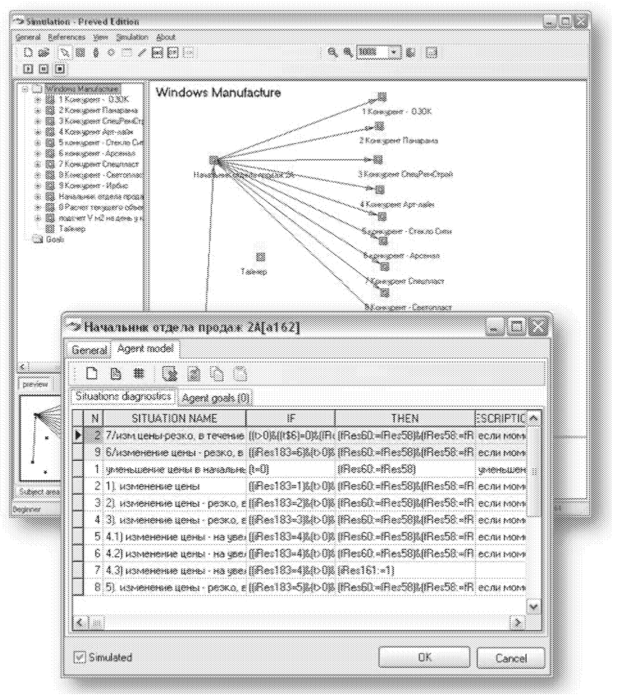
Figure 3: Cellular networks technical economical engineering problem decision search diagram in decision support system BPsim.DSS

Decision support system BPsim.DSS was used on various stages of Ural State Technical University Common Information System (CIS) development and deployment, starting with educational process analysis stage, performing re-engineering, and ending with separate CIS units deployment efficiency estimation.

Model of an agent (decision making person), controlling software development process in Ural State Technical University, was developed in decision support system BPsim.DSS. Model consists of simulation model "Educational process software development" and decision support models, including the main model "CIS implementation options selection". Model knowledge base contains information on networking, hardware and software, information systems, IT-projects, teams of IT-specialists.

Expert system module is used for project alternatives and effective alternative search algorithms knowledge base development. Simulation model is used for separate project stages monitoring, detection of errors and conflicts, occurred on initial planning stage, solution of vis major, that happens during development project control and CIS deployment. Simulation model is based on Spiral model of software lifecycle and is designed in BPsim.DSS.

BPsim.DSS was used for multi-agent dynamic model development of Urals Industrial Group, CJSC (further referenced as UIG). The main reason for modeling is UIG behavior algorithm and pricing strategy development, targeting share of the market growth and transition to higher technological level, increasing enterprise competitiveness. Fragment of the model together with single agent's knowledge base in If-Then form is presented on Figure 4. Decision support system visual output mechanism builder, based on decision search diagrams (Figure 3), as well represents agent knowledge base, based on frame-concepts. So, agent knowledge base may be defined in two ways: productive (Figure 4) and frame-concept – based.



| N | SITUATION NAME | IF | THEN | ESCRIPTIC |
|---|---|---|---|---|
| 2 | 7/изм цены резко, в течение | ((t>0)&((t$6)=0)&(fR | (fRes60:=fRes58)&(fRes58:=fR | если мом |
| 9 | 6/изменение цены - резко, в | ((fRes183=6)&(t>0)& | (fRes60:=fRes58)&(fRes58:=fR | если мом |
| 1 | уменьшение цены в начальн | (t=0) | (fRes60:=fRes58) | уменьшен |
| 2 | 1). изменение цены | ((fRes183=1)&(t>0)& | (fRes60:=fRes58)&(fRes58:=fR | если мом |
| 3 | 2). изменение цены - резко, е | ((fRes183=2)&(t>0)& | (fRes60:=fRes58)&(fRes58:=fR | если мом |
| 4 | 3). изменение цены - резко, е | ((fRes183=3)&(t>0)& | (fRes60:=fRes58)&(fRes58:=fR | если мом |
| 5 | 4.1) изменение цены - на уве | ((fRes183=4)&(t>0)& | (fRes60:=fRes58)&(fRes58:=fR | если мом |
| 6 | 4.2) изменение цены - на уве | ((fRes183=4)&(t>0)& | (fRes60:=fRes58)&(fRes58:=fR | если мом |
| 7 | 4.3) изменение цены - на уве | ((fRes183=4)&(t>0)& | (fRes161:=1) | |
| 8 | 5). изменение цены - резко, е | ((fRes183=5)&(t>0)& | (fRes60:=fRes58)&(fRes58:=fR | если мом |

Figure 4: UIG, CJSC main process and agent's knowledge base

Another application of BPsim.DSS included multi-service telecommunication network models design and telecommunication services area business processes dynamic simulation.

Currently carriers' development departments use their own experimental knowledge base when engineering data-communication networks, while data-communication implementation engineering solutions are foisted by hardware vendors. No operator either makes use of data-communication networks automated design aids, or models various designed/existing network behavior situations when developing new regions.

Development of automated design and modeling methods and aids requires large quantity of primary data for qualitative multi-service network (MSN) technical and economical engineering, which includes: telecommunication hardware and technologies types and parameters; engineers, economists, project managers, marketers, and lawyers' level of knowledge.

Decision support systems fit most for MSN technical and economical engineering problem solution. Decision support systems can make use of simulation, expert and situational modeling (Aksyonov and Goncharova 2006). Decision support systems development and deployment within cellular communication operators is a pressing and needed problem.

The following mathematical methods are used in MSN and business processes modeling, analysis and synthesis tasks: teletraffic theory may be used on all MSN levels except services level; simulation, situational and expert modeling methods are used for business processes analysis and synthesis tasks. Expert and situational modeling methods, neural networks, multi-agent and evolutionary modeling methods can be used in RCP formalization.

Multi-agent resource conversion processes theory is applied for MSN definition from decision support point of view.

Frame-concept and conceptual graphs based approach, offered by A. N. Shvetsov and implemented in form of «Frame systems constructor» expert system shell (FSC), is used as a means of knowledge formalization. A frame-based semantic network, representing feasible relations between frame-concepts, is defined in form of extended UML classes diagram, at the stage of system analysis.

UML sequence diagram is used for visual FSC output mechanism builder implementation. This approach allows visual (in form of flowchart) problem solution flow definition, when solution turns into a sequence of procedure (method/daemon) calls from one frame to another. Hereby, this approach allowed visual object-oriented ontology and knowledge-based output mechanism constructor implementation in form of decision search diagrams.

This constructor, provided that being filled with MSN subject area knowledge and technical and economical engineering rules, represents an intelligent MSN automated engineering system (Intelligent MSN CAD).

Transition from engineering to simulation modeling is implemented by semantic match making between FSC and multi-agent RCP elements.

Intelligent MSN automated engineering and modeling system BPsim.DSS is used for MSN technical and economical engineering problem solution.

## CONCLUSION

Currently AS-TO-BE model data is implemented in CIS program modules and deployed in Ural State Technical University. Due to "Contingent traffic" process improvement and automation dean's office employees work efficiency was raised by 27%, student desk employees work efficiency was raised by 229%. Deployment economical effect is estimated by about 25 thousand euro per calendar year. Economical effect is achieved in shortening and automation of unnecessary document processing stages, information double input prevention and employee load decrease.

Deployment in Urals Industrial Group focused on effective pricing strategy search, considering passive and active competitors behavior. After a series of experiments a pricing policy, resulting in share of the market growth from 6.6% to 20-22%, was determined. Limiting to current problem the optimal values of processes characteristics were calculated. The projected saving rate from the modeling results implementation is estimated by €1.9 million per year.

Simulation, expert, situational and multi-agent modeling integration with object-oriented approach allowed implementation of new object-oriented multi-agent resource conversion processes simulation and decision support method, reflected in development of object-oriented decision support system BPsim.DSS, deployed at companies in Ural region of Russia. Finally, BPsim.DSS system was applied to multi-service network process formalization, model design and decision support in MSN engineering.

## REFERENCES

Aksyonov, K.A. and N.V.Goncharova. 2006. "*Multi-agent resource conversion processes dynamic simulation*", Ekaterinburg, USTU, 311 pages.

Aksyonov, K.A.; E. A. Bykov; E. F. Smoliy; and A. A. Khrenov. 2008. "Industrial enterprises business processes simulateon with BPsim.MAS", *Proceedings of the 2008 Winter Simulation Conference,* Pages 1669-1677.

Aksyonov, K.A.; I.I.Sholina; and E.M.Sufrygina. 2009. "Multi-agent resource conversion process object-oriented simulation and decision support system development and application", *Scientific and technical bulletin, vol. 3 (80). Informatics. Telecommunication. Control.* St.Petersburg, pp.87-96.

Andreichikov, A.V. and O.N.Andreichikova. 2004. "*Intellectual information systems*", Moscow: Finance and statistics, 424p.

Avramchuk, E.F.; A.A.Vavilov; and S.V.Emelianov. 1988. "*Technology of system simulation*", M.: machine construction industry; Berlin: Techniques, 520 p.

Jackson, P. 1998. *Introduction to expert systems*. 3rd Edition. Addison Wesley, 560 pages.

Minsky M. 1975. *A framework for representing knowledge in the psychology of computer vision,*P.H.Winston ed, McGraw-Hill.

Muller, J.P. and M.Pischel. 1993. "*The Agent Architecture InteRRap: Concept and Application*". German Research Center for Artificial Intelligence (DFKI)

Shvetsov, A.N.. 2004. "*Corporate intellectual decision support systems design models and methods*". DPhil research paper. St.Petersburg, Russia, p.461.

Sowa, J.F. 2000. *Knowledge representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove, CA: Brooks/Cole Publishing Co., 594 pages.

# DECISION SUPPORT SYSTEM FOR A REGIONAL SPREADING OF A/H1N1 INFLUENZA VIRUS

David Hill, Romain Barraud, Benoit Crozat, Luc Touraille
UMR CNRS 6158 LIMOS - Blaise Pascal University
Clermont Université – BP 10125
AUBIERE – FRANCE
David.Hill@univ-bpclermont.fr

Alexandre Muzy
UMR CNRS 6240 LISA,
Università di Corsica - Pasquale Paoli,
22, av. Jean Nicoli, 20250 CORTI – FRANCE
A.Muzy@univ-corse.fr

Frederic Leccia
Union Régionale des Médecins Libéraux de Corse
Villa Mérimée - 9 Cours Granval
20 000 AJACCIO – FRANCE

**KEYWORDS**

Epidemic model, A/H1N1, GIS, MAS, Decision Support System, Simulation

## ABSTRACT

The evaluation of the impact of the recent inter-human A/H1N1 influenza spreading on a regional health system has been achieved using a Decision Support System. The latter includes two models: first, a stochastic model composed of a multi-agents system linked to a Geographical Information System; second, a deterministic model using the main parameters of a virus spreading. Both models have been designed in collaboration with doctors (of medicine?). The simulation software has been tested with the data available for Corsica Island and the results have been presented to the local authorities. Small experimental plans, with the main spreading parameters, help in predicting various indicators for the regional health system, such as the number of medical consultations, the number of doctors needed in the different medical areas… The final aim of this support system is for it to be used for local public health decisions, and possibly be adapted to other regions.

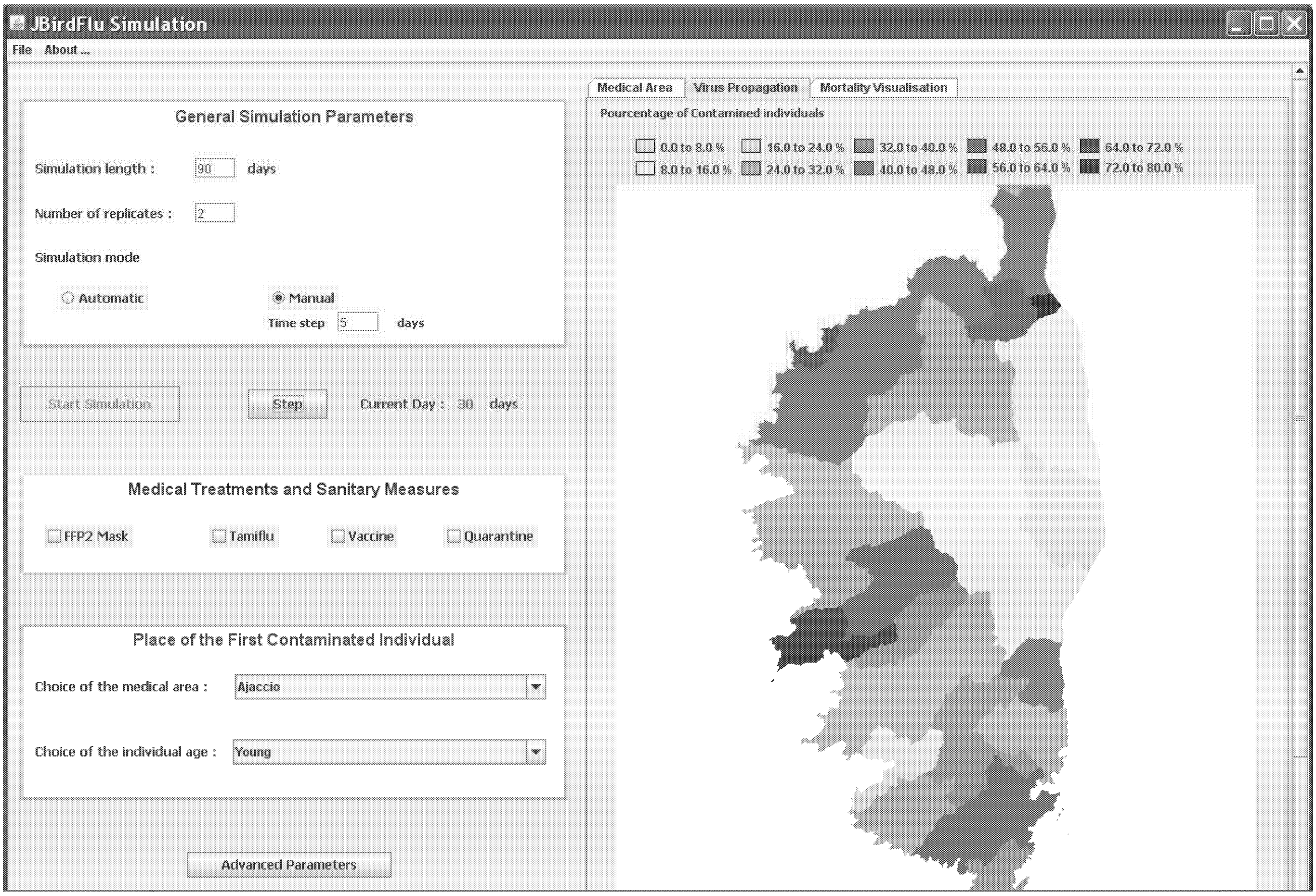
## INTRUDUCTION

Since 2005, the probability of a new influenza pandemic virus was close to 1. In order to be reactive enough, we started to work on decision-aid software able to study the inter-human spreading of the H5N1 strain. Indeed, at that time, Asian countries suggested a probable person-to-person transmission of avian influenza H5N1 (Ungchusak et al. 2005). In 2008, a Chinese group even presented a probable case of inter-human contamination (Wang et al. 2008). In the meantime we were able to propose simulation software, designed by a multi-disciplinary team, including doctors in computer technology (Hill et al. 2008). The simulation was an individual based model, using current data on the existing medical areas of Corsica. The proposed software was dedicated to the H5N1 strain, and combined a Geographical Information Systems (GIS) and a Multi-Agent System (MAS) with a small world like communication between agents. This model could be viewed as a self-organizing system with an important focus on spatial distribution as a key point for the decision aspects. However, in the proposed individual model, the main purpose was to provide indicators for the health system, thus emergent properties were not looked for and the system was too simple to look for dynamic structures at runtime.

After this first software development, we wanted to propose a more generic application. First, we had enough information to customize the software for any kind of virus (including sanitary measures and barriers). Second, we wanted to provide an easy and fast entry point for any region with spreadsheet data as input file in addition to the geographical information system link. Third, we wanted to offer the opportunity to include a comparison of the simulation results obtained with the spatial multi-agent simulation with a more classical deterministic model for pandemic spreading. With this last point we wanted to be able to launch a basic experimental plan with a variation of the pandemic factors (mainly the virus attack rate). Indeed, since the beginning of computer simulation, experimental design has been an active research field for improving the effectiveness of simulated systems (Kempthorne 1952) (Zeigler 1976; Kleijnen 1987; Balci and Sargent 1989).

In the first semester of year 2009, the emergence of A/H1N1 surprised many specialists (Linea 2009) and the question of a pandemic was raised (Sim and Mackie 2009). We are now certain of the pandemic but of course the future virulence and evolution is still unknown. When many factors are imprecise, simulation is a good exercise. Experimental design is then a valuable technique in the decision-making context (Kleijnen and Groenendaal 1992; Hill 1996; Amblard et al. 2003).

Since 2004, many models have been built with the avian flu in mind (Longini et al. 2004, 2005) (Fergusson et al. 2005) (Carrat et al. 2006) (Colizza et al. 2007) (Hill et al. 2008) (Das et al. 2008) and (Iwami et al. 2009). Regarding A/H5N1, model proposals are of course currently limited. A markovian model has been proposed for parameter estimation (Ross et al. 2009). The latter has been used for infectious data from an outbreak of 'Russian influenza' (A/USSR/1977 H1N1) in an educational institution. More recently, a generic serious game has been released; it has been developed by the medical center of the Erasmus University (Rotterdam) and the Ranj Serious Games company (it is even possible to play online: http://www.thegreatflu.com/). Concerning our proposal, the inclusion of two simulation models (a classical deterministic model and a spatial stochastic multi-agents simulation) sharing a common set of parameters led to the design of a Decision Support System (DSS hereafter) including the main sanitary measures (vaccination, prophylaxis, masks and quarantine enforcement).



General Simulation Parameters

Simulation length : 90 days

Number of replicates : 2

Simulation mode

○ Automatic ● Manual

Time step 5 days

Current Day : 30 days

Medical Treatments and Sanitary Measures

□ FFP2 Mask □ Tamiflu □ Vaccine □ Quarantine

Place of the First Contaminated Individual

Choice of the medical area : Ajaccio

Choice of the individual age : Young

Figure 1: Graphical Result of the Interface with the Local Geographical Information System - Spatial Stochastic Simulation (Using a Multi-Agent System with A 25% Attack Rate as Main Parameter)

## DESIGN OF THE DECISION SUPPORT SYSTEM

Figure 1 above presents a sample output of the interactive simulation software we developed in 2008. As mentioned in the introduction, this software was a stochastic multi-agents system connected with a Geographical information system, it is fully described in (Hill et al. 2008). The development was done using an object-oriented analysis and design with a Java implementation. We wanted to enhance this first modular software to obtain a more complete DSS intended to help decision makers in case of intensive flu virus spreading. Our system is a Model-driven DSS, compiling useful information and parameters provided by users to assist decision makers in analyzing a pandemic situation. Using two different models and personal knowledge, this tool will help identifying potential problems and thus participate in the local decision process (support is provided for medical groups, medical organizations or local medical authorities). As in many DSS we have 3 main components:

1. The data base, which includes geographical data, but also territorial and medical data.
2. The model (2 models in this DSS: the stochastic spatial multi-agent system and the deterministic model.
3. The user interface enabling doctors and specialists to run simulations and sets of experiments, thereby helping them in the decision process.

Despite the fact that one of our models uses agents representing individuals with social relationships, we do not deal with cognitive decision-making functions, i.e. we have not introduced artificial intelligence in our DSS – it would have been called an Intelligent Decision Support System (IDSS). The main UML classes of our DSS are presented in figure 2. The data base is composed of four elements: maps, statistic, territory and parameter. These are used by the simulator, which is in charge of handling the model component. Finally, the interactions with the user are obtained through the GUI, configuration and test classes.

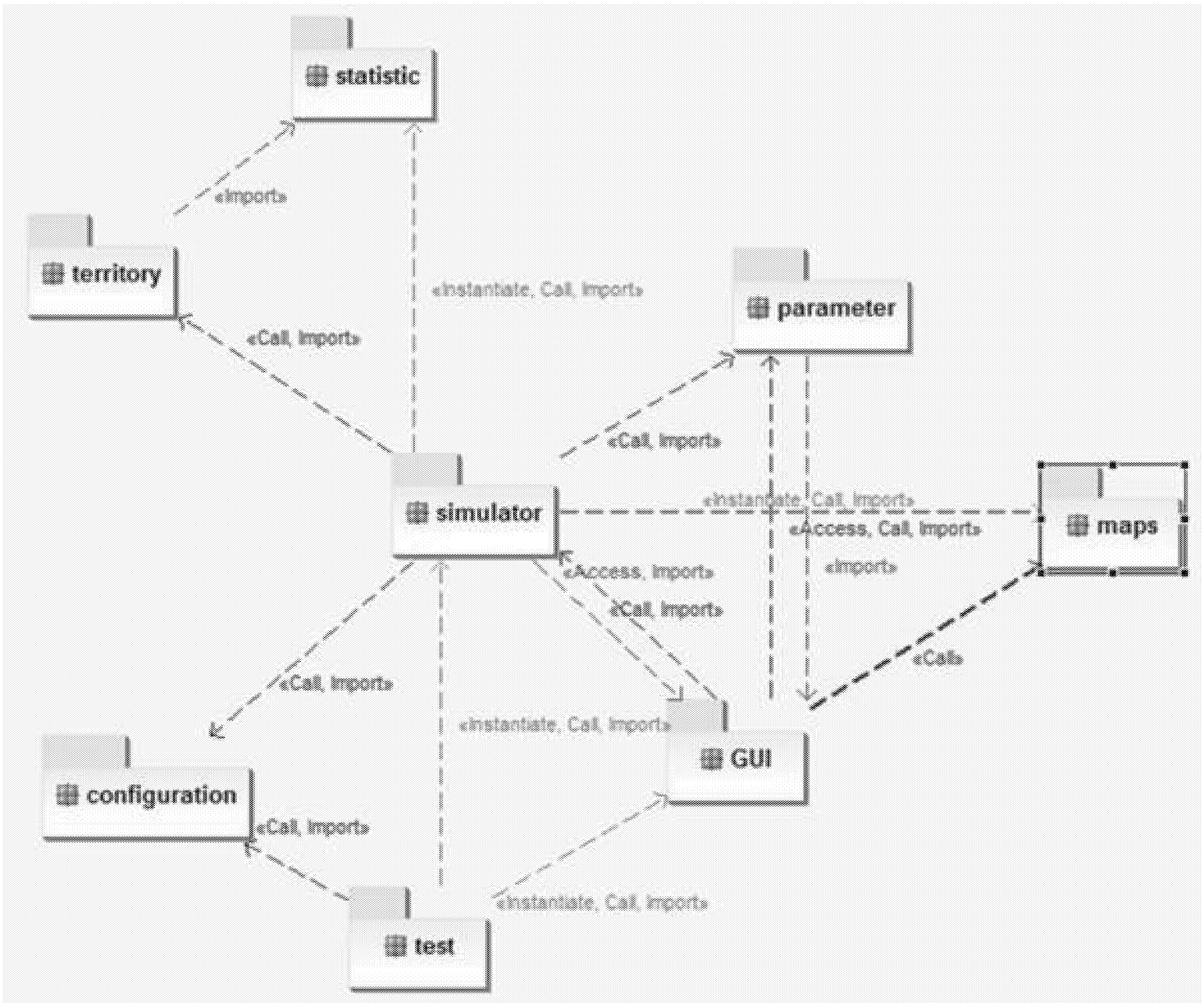


Figure 2: UML Diagram of the Main Classes (Generated from the DSS Software with eUML2)

## INPUT DATA AND PARAMETERS AND DETERMINISTIC MODEL

Figure 1 showed that we have different local maps with contour for the different medical areas of Corsica. In addition, among a set of raw data we also have at our disposal for each medical area:

- The number of doctors and nurses.
- The number of inhabitants – with the number of individuals for 3 age groups: junior (below 20 years old), adult (between 20 and 60 years old) and senior.

Regarding the models parameters, both kinds of model share an XML file mapped with the XStream library to instantiate the parameter class. The file structure is as follows: the first parameter is the attack rate: <attackRate>0.30</attackRate>

Then, the file is split in two parts, the first one describing the characteristics of the first wave of a pandemic virus and the second one dealing with the second wave, statistically more serious than the first wave (we have not considered more than 2 waves).

```
<firstWave>
    <ratioVirus>
      <double>0.4</double>  // Junior factor
   <double>0.5</double>  // Adult factor
   <double>0.1</double>  // Senior factor
    </ratioVirus>
    <ratioConsultation>
   <double>3.0</double>  // Junior
   <double>3.0</double>  // Adult
   <double>3.0</double>  // Senior
  </ratioConsultation>
    …
</firstWave>
```

With this parameter set we specify for each age group: its sensitivity to the virus, the number of consultations if an individual has contracted the virus, the probability of a hospitalization and the death probability in case of hospitalization. This set of parameters is given twice, for the first and second wave.

With this approach the deterministic model can easily compute how many people are concerned in each age group for both waves. To obtain the number of persons infected in a medical area, we simply apply the attack rate to the number of adults present in a medical area. Then the reduction factor of its age group applies (modeling the resistance), it is thus easy to specify that adults below 60 years old are more susceptible to be infected by the virus (which is a common signature for pandemic viruses). For each infected person we apply the mean consultation factor to obtain a number of medical consultations imposed by the new virus. Knowing the number of doctors in a medical area and its average number of consultations, we can determine if we need to reassign doctors in areas poorly equipped in human medical resources.

For the spatial stochastic model we have many specific parameters such a contact matrix depending on the age group (implementing a contact network), the incubation duration time (between 2 and 7 days in the case of H1N1), and so on.

An advanced parameter window (figure 4) enables the testing of sanitary measures: pre-vaccine and pandemic vaccination, antiviral drugs (prophylaxis), masks (chirurgical and FFP2). The quarantine is also considered.

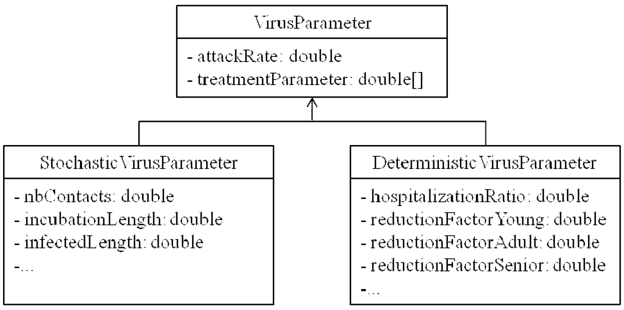
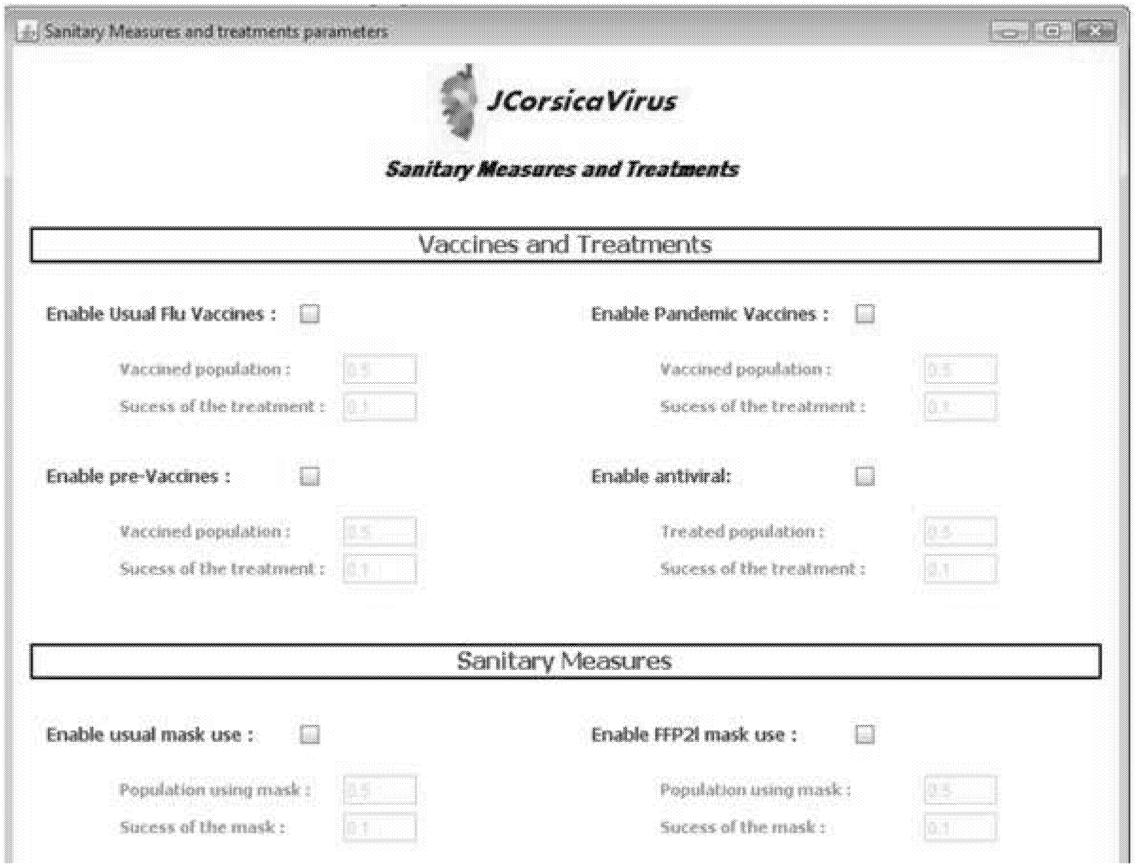

VirusParameter
- attackRate: double
- treatmentParameter: double[]
StochasticVirusParameter
- nbContacts: double
- incubationLength: double
- infectedLength: double
-...
DeterministicVirusParameter
- hospitalizationRatio: double
- reductionFactorYoung: double
- reductionFactorAdult: double
- reductionFactorSenior: double
-...

Figure 3: Subclassing of the Main Parameter Class to Facilitate the Handling of Both Model Types



Sanitary Measures and Treatments

Vaccines and Treatments

Enable Usual Flu Vaccines : □
Vaccined population : 0.5
Sucess of the treatment : 0.1

Enable Pandemic Vaccines : □
Vaccined population : 0.5
Sucess of the treatment : 0.1

Enable pre-Vaccines : □
Vaccined population : 0.5
Sucess of the treatment : 0.1

Enable antiviral: □
Treated population : 0.5
Sucess of the treatment : 0.1

Sanitary Measures

Enable usual mask use : □
Population using mask : 0.5
Sucess of the mask : 0.1

Enable FFP2I mask use : □
Population using mask : 0.5
Sucess of the mask : 0.1

Figure 4: Parameter Window for Sanitary Measures Including: Pre-vaccines, Pandemic Vaccine, Masks (Chirurgical Masks and FFP2 Masks - and Quarantine Enforcement is also Considered but only in the Spatial Stochastic Model

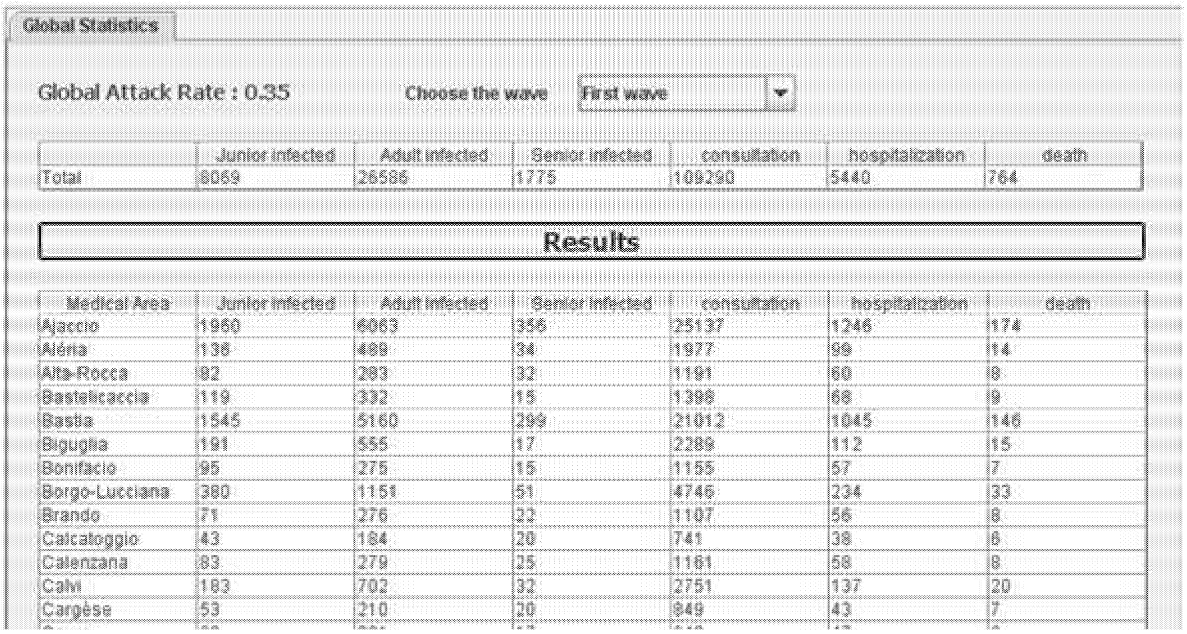
## TEST AND RESULTS FOR CORSICA

Our approach consists of three interdependent levels:

1. The Modeling level: This is the longest phase (following an analysis, design, verification and validation cycle.) Through a deterministic global simulator: modelers evaluate the impact of *global parameters* (attack rate and sanitary measures) on the number of infected people. Through an individual-based stochastic simulator: They evaluate the impact of *indivual-based parameters* (social network, latent and infection periods) on the number of infected

people. Finally, a comparison between both models allows calibrating both global and local factors.

2. Engineering level: Results of the modelling level (e.g., the number of consultations) are compared with resources (e.g., consultations achievable by doctors.)
3. Decision level: Depending on the result of the comparison of both usage and availability of resources, these resources (doctors) can be reallocated in local medical areas.



Global Attack Rate : 0.35

Choose the wave: First wave

| | Junior infected | Adult infected | Senior infected | consultation | hospitalization | death |
|---|---|---|---|---|---|---|
| Total | 8069 | 26586 | 1775 | 109290 | 5440 | 764 |

Results

| Medical Area | Junior infected | Adult infected | Senior infected | consultation | hospitalization | death |
|---|---|---|---|---|---|---|
| Ajaccio | 1960 | 6063 | 356 | 25137 | 1246 | 174 |
| Aléria | 136 | 489 | 34 | 1977 | 99 | 14 |
| Alta-Rocca | 82 | 283 | 32 | 1191 | 60 | 8 |
| Bastelicaccia | 119 | 332 | 15 | 1398 | 68 | 9 |
| Bastia | 1545 | 5160 | 299 | 21012 | 1045 | 146 |
| Biguglia | 191 | 555 | 17 | 2289 | 112 | 15 |
| Bonifacio | 95 | 275 | 15 | 1155 | 57 | 7 |
| Borgo-Lucciana | 380 | 1151 | 51 | 4746 | 234 | 33 |
| Brando | 71 | 276 | 22 | 1107 | 56 | 8 |
| Calcatoggio | 43 | 184 | 20 | 741 | 38 | 6 |
| Calenzana | 83 | 279 | 25 | 1161 | 58 | 8 |
| Calvi | 183 | 702 | 32 | 2751 | 137 | 20 |
| Cargèse | 53 | 210 | 20 | 849 | 43 | 7 |

Figure 5: Results of the Deterministic Model for all the Medical areas in Corsica with a Unique Attack Rate (35%)

### The modelling level

Using the deterministic model, the main parameter is the attack rate. It was easy to run quasi-instantly many scenarios. Figure 5 above presents a simulation for all the medical areas of Corsica (with an attack rate at 35% and a very strong lethality of 2% for a first wave, which is considered as an extremely severe case (almost like the Spanish Flu). Fortunately, the A/H1N1 virus is currently much less virulent, but we have to test the worst cases to determine if the local medical system can cope with the pandemic.

During the testing phase of the spatial stochastic model, we realized that the individual-based parameters (number of contacts by age categories, latent and infection periods, etc.) as well as the duration of infection largely influence the results.

Although the deterministic model provides quickly exploitable results, the spreading is more realistic in the spatial stochastic simulation. In the latter, we can really and finely observe the difference between attack rates and of course the spatial impact is much more evident (cf., Figure 6). For instance, we can notice that some medical areas are less affected by the virus (in particular in the centre of Corsica and in areas with fewer inhabitants). Indeed the population is much less concentrated and the population density is also a factor used in our multi-agent system in addition to the connection matrix.

Finally, a calibration of stochastic parameters (cf. Figure 7) can be achieved by comparing both stochastic and deterministic results.

For both models, the most effective medical measures are the wearing of masks as well as the setting in quarantine. Indeed, with sanitary measures, it is possible to strongly limit the propagation speed of the disease.

### The engineering level

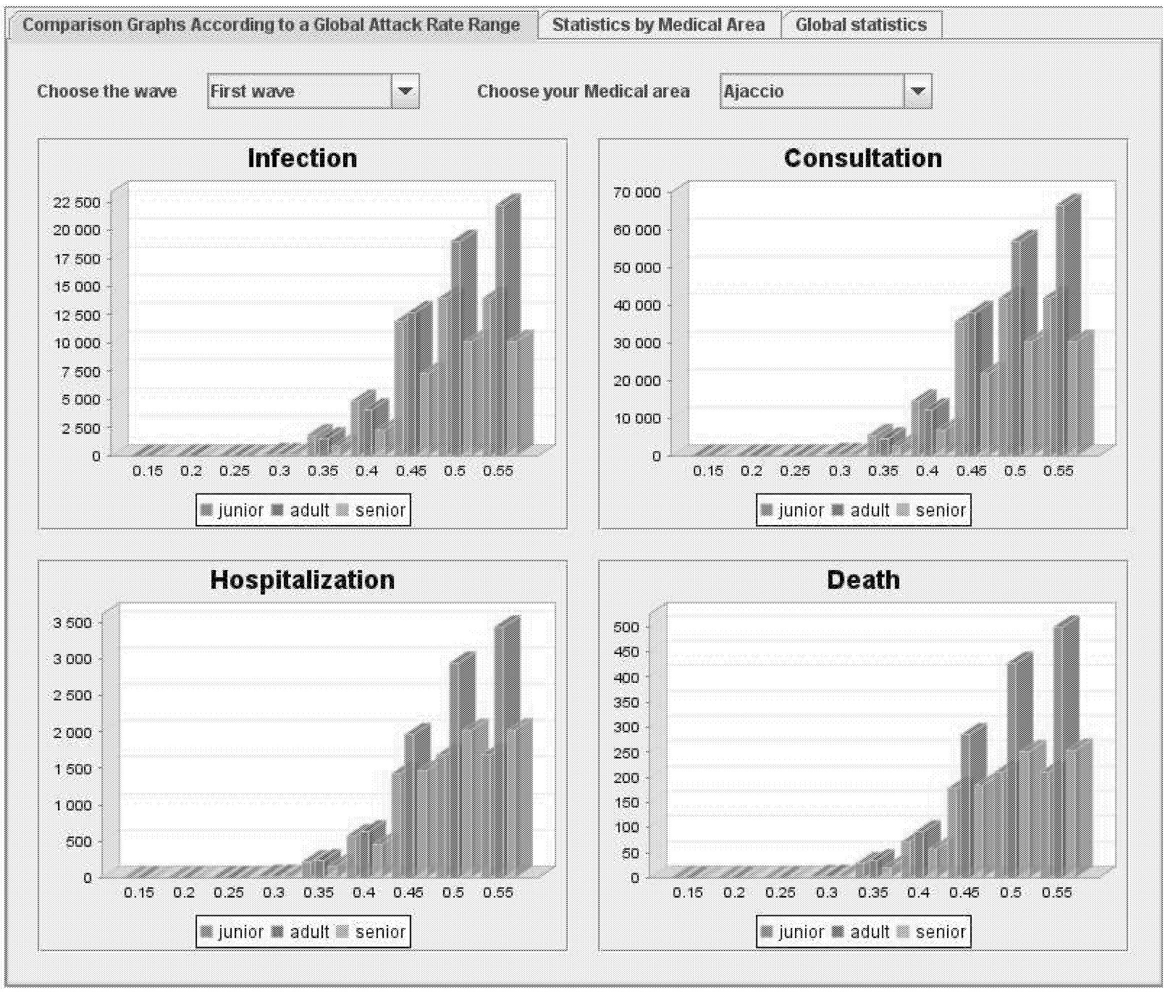
The number of consultations, hospitalizations and deaths are used as indicators and do not have to be considered as prediction. However, they can be used to test probable propagation scenarios, and thus evaluate the safeness of the healthcare network.

Figure 8 depicts such a comparison. *Required consultations* by infected people, *consultations achieved* by doctors, and *consultation margins* (i.e., the number of *Required consultations* minus the number of *consultations achieved*) are depicted. We can notice that for attack rates above 45%, the consultation margin is negative. That is to say that the number of doctors assigned to this medical area is not sufficient.

### Decision level

Using the knowledge acquired on the previous engineering level, new policies can be developed. For example, the consultation margin can be carefully noticed in medical areas. For medical areas with a large consultation margin, we can consider the reassignment of some cases to other medical areas with a poor or negative margin.

According to the number of infected persons, the impact on society organization can also be inferred. Hence, the number of infected adults corresponds approximately to the number of work stoppages (subtracting naturally the number of people who do not work). New organization policies can then be developed for maintaining a minimum activity level for administrations, businesses and industries.



Choose the wave: First wave

Choose your Medical area: Ajaccio

Figure 6: Results of the Spatial Stochastic Simulation with a Simple Experimental Plan (with an Attack Rate Varying from 15 % to 55 % in the Ajaccio Medical Area)

## DISCUSSION AND CONCLUSION

The main goal of our Decision Support System was to provide the user with a modular tool, enabling him to specify the main parameters influencing the propagation of a pandemic on the island of Corsica. It is a classical application of DSS without particular technical innovations. The object-oriented software proposed uses text configuration files (in XML) and a graphical interface to test the results of two models. The first one is a spatial stochastic multi-agent system (individual based) linked to a Geographical Information System providing maps and valuable data for all the Corsican medical areas (Hill et al. 2008). The second simulation model is deterministic and applies the main factors of flu propagation to all the medical areas. A command line version also exists and enables the use of parallel computing – mainly on computing clusters or small SMPs – for further statistical studies following specific design of experiments. For the stochastic multi-agents simulation, we can also specify scenarios of propagation with ASCII files (mainly to test the impact of various occurrence of flu in the different medical areas). A confidence interval is of course computed when we run the stochastic model.

Even if we use this kind of statistical technique, no one can propose accurate predictions for a pandemic flu. The main problem of simulation software for prediction is the lack of reliable statistics. This point has been strongly underlined by the World Health Organization. This does not mean that decision support systems are useless, but our point of view is that we need to run experimental plans with a wide range of values for the most sensitive parameters such as the attack rate (or R0 for the stochastic multi-agent model) and the lethality factor. The latter, for instance, ranges from 0,002% to 0,7% if we observe the current data with more than thousands of cases from different regions of the World. Therefore, all the speculative estimates can only serve to test scenarios. The variation of the main model factor (virus attack rate) has even been proposed directly in the decision support software without having to run another dedicated statistical tool

For the sake of generality, we need to leave aside useful information about some medical areas, which could increase the accuracy of the simulation results. For instance, the presence of a local University in the centre of Corsica (Corti) and the fact that many young students plus teachers and scientists are travelling from Bastia or Aiacciu to Corti will make a significant change in the results of the corresponding medical area. The closure of the university could also be considered. Without data about schools we have not introduced this parameter, which, like a kind of quarantine, will have a significant impact over the speed of virus spreading.

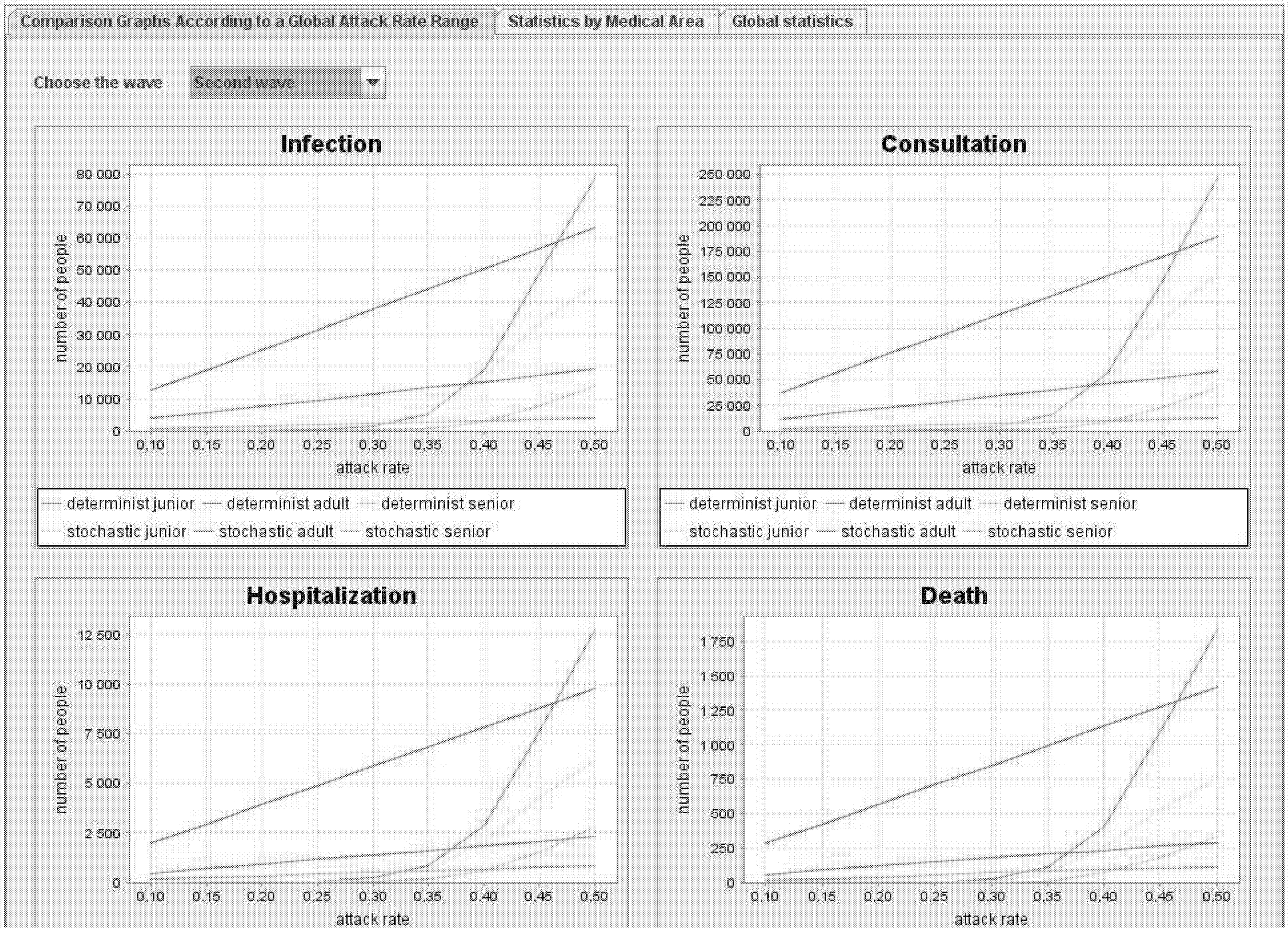
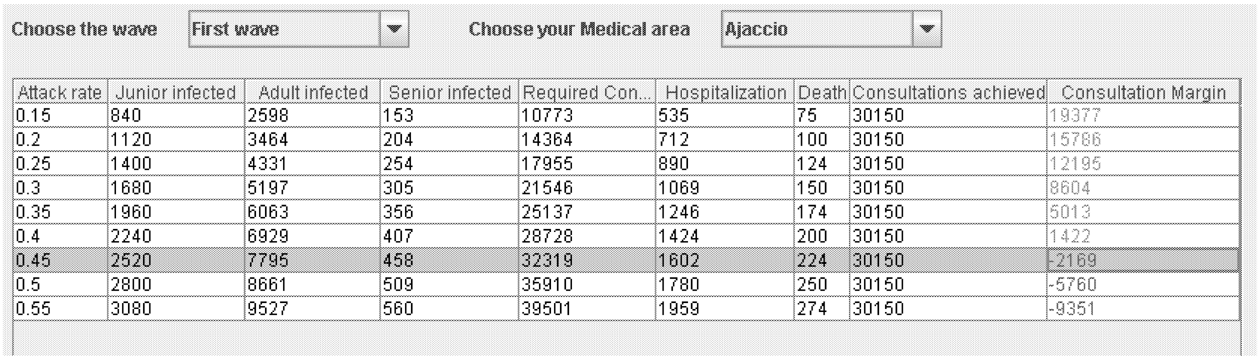


Figure 7: Comparison of Results of the Spatial Stochastic Simulation with the Deterministic Model for a Second Wave of the Virus Spreading over the whole Corsican Population (Attack Rate Varying from 10 % to 50 %)
In the Case of the Stochastic Model, the Attack Rate Corresponds to the Probability of Infection per Person



| Attack rate | Junior infected | Adult infected | Senior infected | Required Con... | Hospitalization | Death | Consultations achieved | Consultation Margin |
|---|---|---|---|---|---|---|---|---|
| 0.15 | 840 | 2598 | 153 | 10773 | 535 | 75 | 30150 | 19377 |
| 0.2 | 1120 | 3464 | 204 | 14364 | 712 | 100 | 30150 | 15786 |
| 0.25 | 1400 | 4331 | 254 | 17955 | 890 | 124 | 30150 | 12195 |
| 0.3 | 1680 | 5197 | 305 | 21546 | 1069 | 150 | 30150 | 8604 |
| 0.35 | 1960 | 6063 | 356 | 25137 | 1246 | 174 | 30150 | 5013 |
| 0.4 | 2240 | 6929 | 407 | 28728 | 1424 | 200 | 30150 | 1422 |
| 0.45 | 2520 | 7795 | 458 | 32319 | 1602 | 224 | 30150 | -2169 |
| 0.5 | 2800 | 8661 | 509 | 35910 | 1780 | 250 | 30150 | -5760 |
| 0.55 | 3080 | 9527 | 560 | 39501 | 1959 | 274 | 30150 | -9351 |

Figure 8. Consultation margin for Aiacciu, with attack rates from 15% to 50%, with 3 consultations per contaminated people.

Among the possible evolutions of this decision support system, a major change could be to add a dynamic link to geographical data. Currently, we have a very limited link with geographical data since bitmap data exported from a Geographical System are post-processed by a separate program before being loaded in memory at the beginning of the simulation. Various techniques can be deployed; in the past, we have deployed many (Coquillard et al. 1995) of them but this would imply much more software development not directly connected with the main aim of this decision support system. However, more essential improvements could be made if we could also have reliable data concerning the remaining stockpile of vaccine, of prophylactic (antiviral) drug, as well as vaccine and prophylactic drug administration capacity. As in (Das et al. 2008), it could also be interesting to consider the duration of hospital stay to combine this model output with the regional hospital bed capacities. A variant of the quarantine enforcement can be to consider only the school and university closure. The current computing performances are satisfactory using Java; they have been tested and profiled to work on a regular personal computer. Moreover, we could adapt the input files and parameters to handle larger areas, from countries to a worldwide simulation. Large experimental designs are planned to be run on computing clusters and not on personal computers. Finally, we think that the current decision support system is flexible enough to be adapted to other viruses.

## REFERENCES

Amblard F., Hill D., Bernard S., Truffot J., Deffuant G., "MDA compliant Design of SimExplorer, A Software to handle simulation experimental frameworks" in *Proceedings of SCSC 2003 Summer Simulation Conference*, Montréal, July 20-24, 2003, pp.279-284.

Balci O., Sargent R.E., Guidelines for selecting and using simulation model verification techniques, *Winter Simulation Conference*, 1989, p. 559-568.

Carrat F., Luong J., Lao H., Sallé A., Lajaunie C. and Wackernage H., 2006, "A 'small-world-like' model for comparing interventions aimed at preventing and controlling influenza pandemics". *BMC Medicine*, 4, n°26. An electronic version of this article can be found online at: www.biomedcentral.com/1741-7015/4/26

Coquillard P., Hill D., Gueugnot J., "Simulation d'Ecosystèmes et Systèmes d'Informations Géographiques : une interactivité nécessaire", *Acte de la conférence Nationale des Parcs Naturels de France*, Vol. 18 Janvier, Ecole des Mines de St-Etienne, 1995, pp. 39-46.

Colizza V., Barrat A., Barthelemy M., Valleron A.-J., Vespignani A., 2007, "Modeling the world-wide spread of pandemic influenza: baseline case and containment interventions", *PLoS Medicine*, Vol. 4, No. 1, pp.13-23.

Ferguson N.M., Cummings D.A.T., Cauchemez S., Fraser C., Riley S., Meeyai A., Iamsirithaworn S., Burke D.S., 2005, "Strategies for containing an emerging influenza pandemic in Southeast Asia", *Nature*, Vol. 437, pp. 209-214.

Hill D.R.C., Object-Oriented Analysis and Simulation, Addison-Wesley Longman, 1996, 291p.

Iwami S., Takeuchi Y., Liu X., Avian flu pandemic: Can we prevent it?, *Journal of Theoretical Biology*, Vol. 257, Issue 1, 7 March 2009, pp. 181-190.

Kempthorne O., Design and Analysis of Experiments, John Wiley & Sons, 1952-1960 -2nd Ed., 631 p.

Kleijnen J. Statistical tools for simulation practitioners, Dekker, New York, 1987, 429 p.

Kleijnen J. and Groenendaal W., Simulation. A statistical Perspective, John Wiley & Sons, 1992, 241 p.

Longini Jr., I.M., Halloran, M.E., Nizam, A., Yang, Y., 2004, "Containing Pandemic Influenza with Antiviral Agents", *American Journal of Epidemiology*, 159, No. 7, pp. 623-633.

Longini M., Nizam A., Xu S., Ungchusak K., Hanshaoworakul W., Cummings D.A.T., Halloran M., 2005, "Containing Pandemic Influenza at the Source", *Science*, 2005. July 12th. 42p.

Linea B., Etonnants virus influenza : bilan de deux années de grippe saisonnière et surprises, *Médecine et Maladies Infectieuses*, Vol. 39, Suppl. 1, 10ème Journées Nationales d'Infectiologie, June 2009, Page S6.

Ross J.V., Pagendam D.E., Pollett P.K., On parameter estimation in population models II: Multi-dimensional processes and transient dynamics, *Theoretical Population Biology*, Vol. 75, Issues 2-3, March-May 2009, pp.123-132,

Sim F. and Mackie P., Pandemic or no pandemic: Emergence of swine influenza A (H1N1) in 2009, *Public Health*, Vol. 123, No. 6, June 2009, pp.405-406.

Wang, H., Feng, Z., Shu, Y., Yu, H., Zhou, L., Zu, R., Huai, Y., (...), Wang, Y., "Probable limited person-to-person transmission of highly pathogenic avian influenza A (H5N1) virus in China", *The Lancet*, Vol. 371 (9622), 2008, pp.1427-1434.

Zeigler B.P., Theory of Modeling and Simulation, Wiley Interscience, New York, 1976.

## ACKNOWLEDGEMENT

A specific word of thanks is given to Romain Barraud and Benoît Crozat for their strong involvement in the development of this software, even whilst they were in their last year of internship for the French grade of "Ingénieur" in Computer Science and Modelling at ISIMA (www.isima.fr)

**DAVID R.C. HILL** is currently Vice President of Blaise Pascal University in charge ICT and Director of the local Inter-Univ. Computing Center. Past deputy director of ISIMA Computer Science & Modeling Institute, Professor Hill has authored or co-authored more than a hundred technical papers and journal papers and he has a number of text books (www.isima.fr/~hill)
e-mail : David.Hill@univ-bpclermont.fr

# MEMORY ASSISTANT IN EVERYDAY LIVING

Ângelo Costa[1], Paulo Novais[2], Ricardo Costa[3], José Neves[2]

[1,2]CCTC, Departamento de Informática,
Universidade do Minho, Braga, Portugal
Email: [1]angelogoncalocosta@gmail.com,
[2]{pjon,jneves}@di.uminho.pt

[3]CIICESI, College of Management and
Technology - Polytechnic of Porto, Felgueiras,
Portugal
Email: [3]rcosta@estgf.ipp.pt

## KEYWORDS

e-Health, Personal Memory Assistant, Ambient Assisted Living.

## ABSTRACT

Memory is one's mental ability to store, to retain, and recall information, representing past and future, our dreams and/or expectations. However, as the human been ages, the capacity of remembering decreases as well the ability to pile up new memories, therefore affecting our quality-of-living and lowering our self- esteem. This configures a social and human dilemma. With the present work we intend to address some of these problems, in terms of a Personal Memory Assistant (PMA), in order to help the user to remember things and occurrences, making it in a proactive mode. It will also cater for some form of relaxation on the part of the user.

## INTRODUCTION

An intrinsic characteristic of the human being is his/her memory. Our remembering capacity is of the utmost importance as it gives us the sense of being and the capacity to have a social life, to remember how things are done and to envision the future. This work is oriented to an older population, typically retired, with spare free time, helping them in schedule events with minimal interaction and suggesting activities to fill their unfilled time.

A PMA tool and a Social Enabler one are presented, as well as the simulation results of their impact on the user life.

### Ageing Population

According to the United Nations Population Fund (UNPF), the life expectancy of the world population is increasing and the birth rate of children is decreasing rapidly. The UNPF estimates that the European Population decreased 13% in a fifty year period, increasing the age average to 48 years old (UNFPA 2002) at the age of 50 the human beings are severely affected by it, being the forgetfulness of events, namely the more recent ones, one of the most occurred symptoms. Memory is no more than the concept that refers to the process of remembering (Mohs 2007), aging, especially if associated with chronic diseases, affects our ability to remember. There is still no known way of reversing the human brain loss of information, so a possible solution may be the use of computational systems to store and retrieve all that data. Thru the use of an agenda and/or calendar, we may reach the goals set to this work. However, the current technologies fail in this point, by misinterpretation of the actual needs of the users or the directions taken to approach the problem. On the other hand it has been shown that scheduling and storing intelligently the user's activities, making easier the communication with their peers and relatives, may greatly improve the elderly self-esteem on their daily activities (Aguilar, et al. 2004) (S. J. Brown 2003).

### Memory Assistant

It is still unknown in detail how to store memories in the brain, how it works. At the present moment we can only foresee the way it works in terms of organization and memorization. This leads us to conclude that a human being with loss of memory can be helped by means of PMAs. An area that covers a broad spectrum, with many projects arising with different objectives, although they are scattered and have many different focus, like time-lapse recorders and event managers. In the agenda and organization arena, resulting from a thorough investigation, there are no projects being developed at the present time. There are, however, other projects in the memory assistant arena, which will be object of attention, serving also as a small benchmark of what is being done nowadays. None are directly related with our work, but some ideas can be correlated and some features may be considered.

*HERMES*

HERMES is aimed to provide cognitive care (Jiang, Geven e Zhang 2009). This project is supported by the EU under the Framework Programme 7. It is designed in order to provide an independent living and social participation, improving the quality of living of the user. The main objective is to develop a system, software and hardware, which will reduce the negative impact of declining cognitive capabilities, particularly the memory. An important feature of the this project is the implementation of a monitoring system which should be able to record every action and choice of the user in order to build an association "map", and based on that "map", creating a pattern that emulates the human memory mechanisms. Despite its ambitious goals, HERMES is still in a very early stage of development with an amount of problems that wave no known time of resolution.

*M4L: Memories for Life*

Developed by the Engineering and Physical Sciences Research Council, has as objective to use technological solutions to help the user's memory (P. J. Brown 2004). This project has focus on five different fields: health, private life, education, entertainment and science. Currently it is proposed a raw data archive centre that can store information of different users. The access to the data can be performed by hand-held devices and computers that have constant connection to the server. Basically it aims to save all the user information and put it available in one place. As a result, it intends to eliminate all the paper used and having, at the same time, all the people linked to the system connected. The project is still in a investigation and implementation phase, and any concrete results will be only be achieved in the years ahead.

### VirtualECare Project

Initially developed under the VirtualECare project (Novais, Costa e Carneiro, et al. 2008) the iGenda has become more than a module, as it grew apart as independent project (Figure 1).

The main objective of the VirtualECare project is to build a multi-agent and multi-module system capable of monitoring and interact with its users providing health care services, thus, increasing their quality-of-living. Its distributed architecture is composed of several modules connected through a network, having each one a unique role. It has also become a great cradle of smaller projects that are now, waiting for their spotlight and independency.

## AGENDA SCHEDULING AND ORGANIZATION

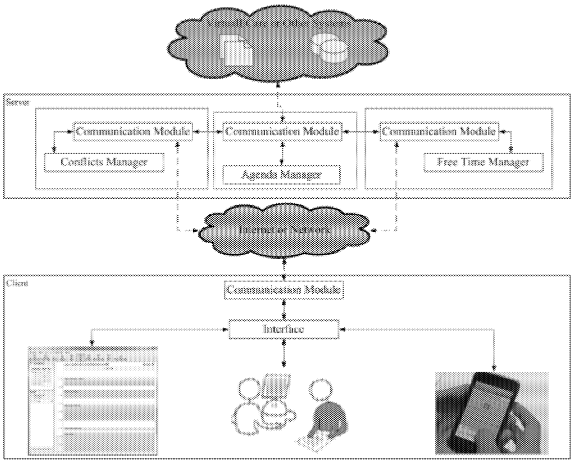


Figure 1: Modules Scheme of iGenda

Our main objective in this work is to produce an intelligent scheduler that interacts with the user through computational means, creating a product that will help the user to remember relevant information or events, i.e., a PMA, by emulating the way the brain processes new events and reorganises the already scheduled ones. For rescheduling an event we normally see in the agenda, which are the events able to be moved to other place(s), considering factors like their relevance and attached problems (e.g. other persons involved, a meeting).

It may help specially the ones with loss of memory, by sustaining all the daily events and making known to the user when it is time to put them into action. It will be able to receive information delivered by any platform and organise it in the most convenient way, according to predefined standards and protocols, so that the user will not need to be bothered about planning or scheduling specific events and tasks. The iGenda is a hierarchy of states and events (Figure 1), intended to deal with the users expectations. It is constituted by the Agenda Manager, the Free Time Manager, the Conflicts Manager and the Interface Manager. All the project modules were written in Java and Prolog, being dependant on the JADE to manage the agents.

### Communication

The communication protocol complies with the FIPA-ACL XML (Caire 2006). All the modules will be compliant with this standard since they are all JADE implemented agents. Messages carry the information of updates and direct announcements. These messages are sent and received through the several modules and clients.

Also with the utilization of JADE it is possible the distribution of agents platforms across machines, which may have different operative systems with the possibility to migrate agents among machines at run-time. It also provides portability, which means that any module may run in different machines, being positioned on any part of the world.

Due to its small size the messages are lightweight, easy to be transmit between agents; due to tolerance to errors a message buffer was also created to support any miscommunication or loss of connection with the agents.

The new event messages will carry in their content XML formatted information (Listing 1).

Listing 1: An example of a new event message

```
(inform
      :sender       Admin@AdmM
      :receiver     AM0001@AmM
      :reply-with   1203302
      :language     XML
      :ontology     new_event
      :content      <new-event>
            <id>    0001</id>
            <prio>  3</prio>
            <sum>   "Visit dentist"</sum>
            <dateb>20090514</dateb>
            <timeb>110000  </timeb>
            <datee>20090514</datee>
            <timee>123000  </timee>
      </new-event>
)
```

### Agenda Manager

The Agenda Manager (AM) sets the bridge between the remaining parts of the manager system and the scheduling one, using the communication infrastructure available to receive and to send requests. As a result, the AM stands for the starting point of all the work that has to be done. It configures a two stage module application. It manages the incoming events to be scheduled and programs the time that triggers the Free Time Manager.

It also supports the reception of multiple messages, thus increasing the overall performance of the system.

Indeed, the AM manages the entire project. Its assessment modifies the way the project works.

### Conflicts Manager

The Conflicts Manager (CM) module is intended to assure that there is no overlap of activities. This module schedules or reorganizes the events that are received from the AM, making sure that they are in accordance with the other events. When a collision of different hierarchical events is detected, the outcome will be decided by methods of intelligent conflict management. In case of overlapping events with the same priority level, the notification of overlapping is reported to the sender, so he/she may try to reschedule to a different time slot.

The CM will work whenever the Agenda Manager considers it is appropriate. This module has also the capacity to manage all the connections with the other users as well as with the user relatives.

The Conflicts Manager operation can be explained in the following way:

1. When an operation is done by an administrator, the AM receives the message and calls the CM.
2. The CM enters in action by reading all the calendar files, parsing the new event and using the CLP engine to compare the priority levels of eventually conflicting events.
3. It is created a new *ICS* calendar ready to be delivered to the user.
4. A new message is then sent to the user and to the administrator, notifying them that a new Calendar is available and that the new insertion was successful.

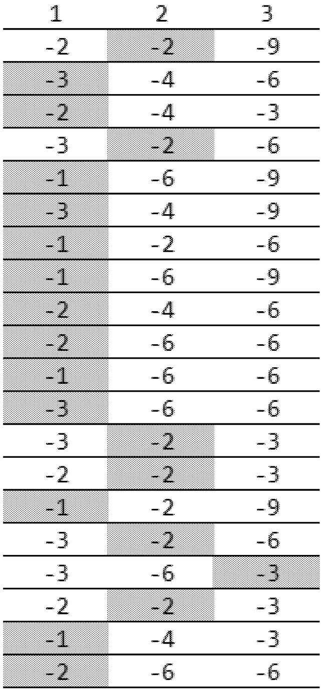
### Free Time Manager

The Free Time Manager (FTM) will fit recreational activities in the free spaces on the user calendar, then trying to enforce the well being of the user. These activities configure an important milestone for an active ageing on the part of the user, once it promotes his/her cultural, educational and conviviality conducts, based on an individual plan. The FTM has a database that contains information of the user's favourite activities, previously checked by the decision support group.

As the project evolved, an idea emerged, that of inspiring the free time activities on the part of the user. One of them was the inclusion of community related projects. These projects were not developed here, but were imported and adjusted to the results expected to be achieved with this work. The two projects considered were the Time Bank (Castolo, Ferrada e Camarinha-Matos 2004) and ePal, which make up a type of community volunteerism.

$$v = -1\, x_{i=0}^{n}\, rand(|n|) \qquad (1)$$

The Free Time Manager uses a distribution function (1); it decides on the activity that is inserted into the user's free time. For instance, in a three activities packet, the rate for the activity of higher priority is of 70%, 25% for the second and 5% for the remaining one. The activities are merely suggestive, it comes to the user to decide to execute them or not. On the other hand the activities chosen are those that fit into the time space that is available.



| 1 | 2 | 3 |
|---|---|---|
| -2 | ▨ -2 | -9 |
| ▨ -3 | -4 | -6 |
| ▨ -2 | -4 | -3 |
| -3 | ▨ -2 | -6 |
| ▨ -1 | -6 | -9 |
| ▨ -3 | -4 | -9 |
| ▨ -1 | -2 | -6 |
| ▨ -1 | -6 | -9 |
| ▨ -2 | -4 | -6 |
| ▨ -2 | -6 | -6 |
| ▨ -1 | -6 | -6 |
| ▨ -3 | -6 | -6 |
| -3 | ▨ -2 | -3 |
| -2 | ▨ -2 | -3 |
| ▨ -1 | -2 | -9 |
| -3 | ▨ -2 | -6 |
| -3 | -6 | ▨ -3 |
| -2 | ▨ -2 | -3 |
| ▨ -1 | -4 | -3 |
| ▨ -2 | -6 | -6 |

Figure 2: An example of function (1) run

In Figure 2 it is depicted an example of the choices made, i.e., in green are posted the winning activities for each round, as well as the values returned by the function's (1) for each activity. As it may be seen the activity 1 presents a rate of choices of 65%, the activity 2 of 30% and the activity 3 of 5%, values that are proximal of the expected ones.

### Interface Manager

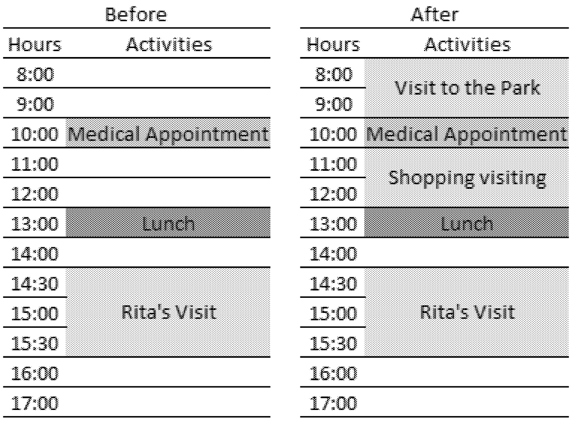
The interface intends to be intuitive and easy to use. It is known that the elderly have some difficulties with the new technologies, so the interface must be intuitive and easy to use. Large buttons are used and only the necessary information is displayed. A variable warning system is also available. When an event is triggered or accomplished, the user is informed.

## SIMULATION

A simulation platform was devised to test the overall system.

### Free Time Manager

A typical result of a run of the FTM module is given in Figure 3. It presents not only a calendar where a set of activities were previously scheduled, but also the results obtained with the FTM module, i.e., a calendar filled with full of fun activities.

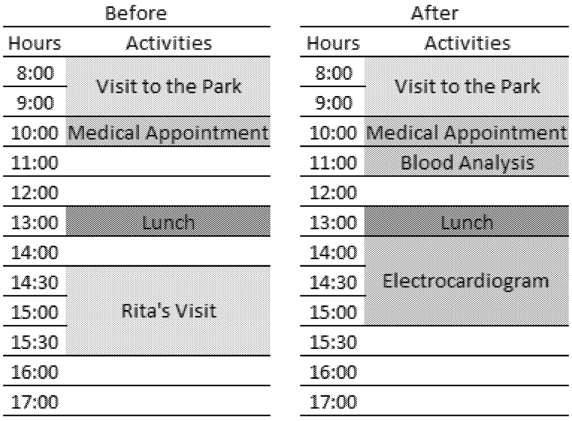


| Before | | After | |
|---|---|---|---|
| Hours | Activities | Hours | Activities |
| 8:00 | | 8:00 | Visit to the Park |
| 9:00 | | 9:00 | |
| 10:00 | Medical Appointment | 10:00 | Medical Appointment |
| 11:00 | | 11:00 | Shopping visiting |
| 12:00 | | 12:00 | |
| 13:00 | Lunch | 13:00 | Lunch |
| 14:00 | | 14:00 | |
| 14:30 | Rita's Visit | 14:30 | Rita's Visit |
| 15:00 | | 15:00 | |
| 15:30 | | 15:30 | |
| 16:00 | | 16:00 | |
| 17:00 | | 17:00 | |

Figure 3: A calendar's before and after picture of a FTM call

All the activities planned came from the database present in the system. On the other hand, the chosen activities have to fit into the time slot available. For example, and as it is shown on the calendar's picture before a FTM, let us consider three events only, i.e., a visit to the doctor, the lunch and the granddaughter visit (Figure 3). A lot of calendar's spaces remain available, and may be occupied by other activities. As it was explained before these activities were selected by the distribution function (1), having into consideration the priority values as also a randomization element that introduces variance to the final result.

**Conflicts Manager**

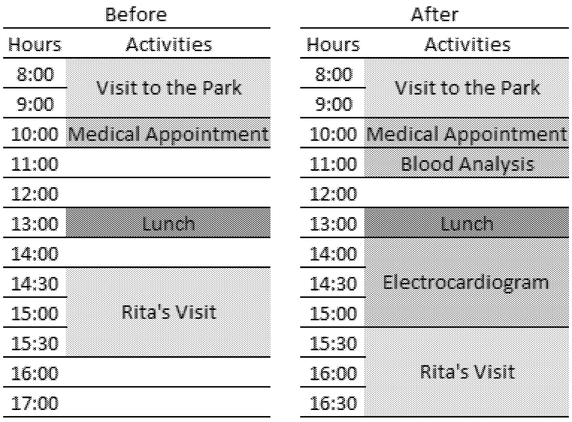


| Before | | After | |
|---|---|---|---|
| Hours | Activities | Hours | Activities |
| 8:00 | Visit to the Park | 8:00 | Visit to the Park |
| 9:00 | | 9:00 | |
| 10:00 | Medical Appointment | 10:00 | Medical Appointment |
| 11:00 | | 11:00 | Blood Analysis |
| 12:00 | | 12:00 | |
| 13:00 | Lunch | 13:00 | Lunch |
| 14:00 | | 14:00 | Electrocardiogram |
| 14:30 | Rita's Visit | 14:30 | |
| 15:00 | | 15:00 | |
| 15:30 | | 15:30 | |
| 16:00 | | 16:00 | |
| 17:00 | | 17:00 | |

Figure 4: The calendar's picture after a direct scheduling without rearrangement

The Figure 4 denotes a direct scheduling. The result points to the elimination of an activity, Rita's Visit, replaced by one of higher priority, a healthiness one.

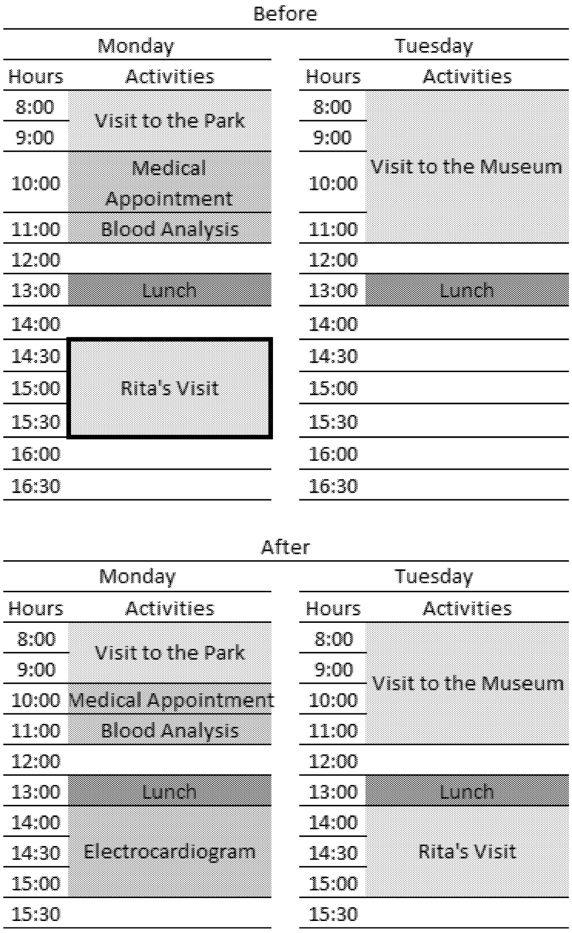
The CM avoids a direct scheduling with no regard for the previous scheduled events. It operates intelligently by triumph over all the problems, i.e., rescheduling the overlapping events. As it is seen in Figure 5, all the events that are healthiness have priority 1. If they overlap with any other activities, with minor priority, the formers are removed, and any person associated with this event is notified of that fact.



| Before | | After | |
|---|---|---|---|
| Hours | Activities | Hours | Activities |
| 8:00 | Visit to the Park | 8:00 | Visit to the Park |
| 9:00 | | 9:00 | |
| 10:00 | Medical Appointment | 10:00 | Medical Appointment |
| 11:00 | | 11:00 | Blood Analysis |
| 12:00 | | 12:00 | |
| 13:00 | Lunch | 13:00 | Lunch |
| 14:00 | | 14:00 | Electrocardiogram |
| 14:30 | Rita's Visit | 14:30 | |
| 15:00 | | 15:00 | |
| 15:30 | | 15:30 | Rita's Visit |
| 16:00 | | 16:00 | |
| 17:00 | | 16:30 | |

Figure 5: The calendar's picture after a CM call

The "Visit to the Park" and "Rita's Visit" where immediately changed as they have a low priority factor. Rita was noticed of the rescheduled event.



Before

| Monday | | Tuesday | |
|---|---|---|---|
| Hours | Activities | Hours | Activities |
| 8:00 | Visit to the Park | 8:00 | Visit to the Museum |
| 9:00 | | 9:00 | |
| 10:00 | Medical Appointment | 10:00 | |
| 11:00 | Blood Analysis | 11:00 | |
| 12:00 | | 12:00 | |
| 13:00 | Lunch | 13:00 | Lunch |
| 14:00 | | 14:00 | |
| 14:30 | Rita's Visit | 14:30 | |
| 15:00 | | 15:00 | |
| 15:30 | | 15:30 | |
| 16:00 | | 16:00 | |
| 16:30 | | 16:30 | |

After

| Monday | | Tuesday | |
|---|---|---|---|
| Hours | Activities | Hours | Activities |
| 8:00 | Visit to the Park | 8:00 | Visit to the Museum |
| 9:00 | | 9:00 | |
| 10:00 | Medical Appointment | 10:00 | |
| 11:00 | Blood Analysis | 11:00 | |
| 12:00 | | 12:00 | |
| 13:00 | Lunch | 13:00 | Lunch |
| 14:00 | Electrocardiogram | 14:00 | Rita's Visit |
| 14:30 | | 14:30 | |
| 15:00 | | 15:00 | |
| 15:30 | | 15:30 | |

Figure 6: The calendar's picture after a CM call in two consecutive days

In Figure 6 it is shown:

- The result of a CM call on a previous event-filled calendar. As it can be seen the "Rita's Visit" is moved to Tuesday. This happened once Rita's visit was not programmed when the iGenda scheduled it. Rita's visit was moved to the day after, as she confirmed in the user calendar that she had that time slot available;

- The result of shifting activities between two consecutive days. In Figure 6 it is shown a calendar's picture after a CM's call that happen in two consecutive days. As it can be seen the Rita's visit is on Tuesday. This happened because Rita could not be present when the iGenda scheduled it. She has moved the event to the next day as she confirmed on the user calendar that she had that time slot available.

## FUTURE WORK

In terms of future work we will consider some ideas that came up and surfaced during this work, namely:

- A Case Based Reasoning model will be implemented, so that the iGenda will have the capacity to remember and learn from past decisions (Aamodt e Plaza 1994);
- A weather detection mechanism will be also fixed, in order to provide iGenda with the possibility to optimize the selection of events ; and
- A Geographic Information System.

## CONCLUSIONS

Although this project has been conceived to set one of the functionalities of the VirtualECare project, it surpasses such endeavour. It turned into an independent and self-sufficient project, with potential to be used in other environments and situations.

Regardless of how it will evolve in the future, there are still problems and critical decisions to be made, namely the "density" problem, where by density we mean overcrowding the calendar of the user with too many activities, making it more stressful than relaxing.

It also makes the difference to other PMAs, once it introduces the component of free time occupation, a problem to be addressed in terms of socialization; i.e., in terms of a process by which the user learn acceptable and unacceptable behaviours for a give environment.

## REFERENCES

Aamodt, Agnar, e Enric Plaza. "Case-based reasoning; Foundational issues, methodological variations, and system approaches." *AI Communications* 7 (1994): 39-59.

Aguilar, José-María, Javier Cantos, Guillermo Expósito, e Pedro Gómez. "Tele-assistance Services to Improve the Quality of Life for Elderly Patients and their Relatives: The Tele-CARE Approach." *The Journal on Information Technology in Healthcare*, 2004.

Brown, Peter J. "GC3: Memory for Life: Getting Things Back." 2004.

Brown, Steve J. "Next generation telecare and its role in primary and community care." *Health \& Social Care in the Community* 11 (2003): 459-462.

Caire, Giovanni. "Using the XMLCodec add-on." *http://jade.tilab.com/doc/tutorials/XMLCodec.html*, 2006.

Castolo, O., F. Ferrada, e L. Camarinha-Matos. "TeleCARE Time Bank: A Virtual Community for Elderly Care Supported by Mobile Agents." *The Journal on Information Technology in Healthcare*, 2004: 119-133.

Novais P., Costa R., Carneiro D., Machado J., Lima L., Neves J., Group Support in Collaborative Networks Organizations for Ambient Assisted Living, in Towards Sustainable Society on Ubiquitous Networks, Makoto Oya, Ryuya Uda, Chizuko Yasunobu (eds), Springer-Verlag, ISBN 978-0-387-85690-2, pp 353-362, 2008.

Jiang, Jianmin, Arjan Geven, e Shaoyan Zhang. "HERMES: A FP7 Funded Project towards Computer-Aided Memory Management Via Intelligent Computations." 2009: 249-253.

Mohs, Richard C. "How Human Memory Works." *http://health.howstuffworks.com/human-memory.htm* HowStuffWorks.com (2007).

UNFPA. "Population Ageing and Development: Operational Challenges In Developing Countries." 2002.

## BIOGRAPHY

**ÂNGELO COSTA** is an MSc student at the Department of Informatics, the University of Minho, Braga, Portugal. His current research directions span the fields of Ambient Intelligence, Collaborative Networks and Multi-agent Systems applied to the Healthcare sector.
angelogoncalocosta@gmail.com

**PAULO NOVAIS** is a Professor of Computer Science at the Department of Informatics at the University of Minho, Braga, Portugal. He received a PhD in Computer Science from the same university in 2003. His current research directions span the fields of Knowledge Representation and Reasoning, Multi-Agent Systems, Ambient Intelligence, Collaborative Networks and AI and The Law. He is supervising several M.Sc. and Ph.D. projects and he has published more than sixty papers in international journals, conferences and workshops and has co-edited two books. He is Vice-president of APPIA, the Portuguese Association for Artificial Intelligence.
pjon@di.uminho.pt

**RICARDO COSTA** is a Professor of Computer Science at the College of Management and Technology of Polytechnic of Porto, Felgueiras, Portugal. He is also a PhD student at the Department of Informatics at the University of Minho, Braga, Portugal. His current research directions span the fields of Ambient Intelligence, Collaborative Networks and Reasoning Systems applied to the Healthcare sector.
rcosta@estgf.ipp.pt

**JOSÉ NEVES** is Full Professor of Computer Science at the Department of Informatics at the University of Minho, Braga, Portugal. He received a PhD in Computer Science from Herriot Watt University, Edinburgh, Scotland, in 1984. His current research directions span the fields of Evolutionary Intelligence, Knowledge Representation and Reasoning, Multi-agent Systems, and AI and The Law. He has published more than one hundred papers in international journals, conferences, workshops and two books.
jneves@di.uminho.pt

# SIMULATION AND AI

# AN ALTERNATIVE MEASUREMENT OF THE ENTROPY

# EVOLUTION OF A GENETIC ALGORITHM

Manuel Cebrian (1), Manuel Alfonseca (2) and Alfonso Ortega (2)
(1) Human Dynamics Lab, The Media Lab, Massachusetts Institute of Technology,
cebrian@media.mit.edu
(2) Escuela Politécnica Superior, Universidad Autónoma de Madrid,
{Manuel.Alfonseca, Alfonso.Ortega}@uam.es

## KEYWORDS

Genetic Algorithms, Entropy, Thermodynamics, Kolmogorov Complexity

## ABSTRACT

In a genetic algorithm, fluctuations of the entropy of a genome over time are interpreted as fluctuations of the information that the genome's organism is storing about its environment, being this reflected in more complex organisms. The computation of this entropy presents technical problems due to the small population sizes used in practice. In this work we propose and test an alternative way of measuring the entropy variation in a population by means of algorithmic information theory, where the entropy variation between two generational steps is the Kolmogorov complexity of the first step conditioned to the second one. We also report experimental differences in entropy evolution between systems in which sexual reproduction is present or absent.

## INTRODUCTION

The evolution over time of the entropy of a genome within a population is currently an interesting problem which is conjectured to be connected to the evolution of the complexity of organisms in a genetic algorithm [Adami et al., 2000, Adami and Cerf, 2000]. The complexity of the genome of an organism is considered to be the amount of information about its environment it stores. That is, evolution would cause the appearance of more complex sequences, which correspond to more complex phenotypes. This hypothesis states that natural selection acts as a Maxwell demon, accepting only those changes which adapt better to the environment and give rise to more complex individuals with genomes of lower entropy. This idea was tested by the simulation of a very simple system of asexual individuals in fixed environmental conditions.

However, it is well known that that the computation of the entropy as

$$\mathrm{H(genome)} = -\sum_{x\in population} \mathrm{p(x)\log p(x)} \quad (1)$$

where x is one of the genomes in the population, has technical complications, due to the large size of the sample needed to estimate it with accuracy [Adami and Cerf, 2000, Herzel et al., 1994, Basharin, 1959]. In practice, it is usually estimated as

$$\mathrm{H(genome)} = \sum_{i=1..size(genome)} \mathrm{H(i)} \qquad (2)$$

i.e., as the sum of the entropy contributions of each locus in the genome. This estimation misses the entropy contributions due to epistatic effects. Some sophisticated statistical methods can be used to remedy this (see Appendix in Adami et al. [2000]), although we will not deal with them in this work.

An still unexplored way to overcome this problem is to estimate the entropy of a genome as its average Kolmogorov complexity

$$\mathrm{<K(genome)>_{genomes\ in\ the\ population}} \rightarrow \mathrm{H(genome)} \quad (3)$$

(see Cover and Thomas [1991], Kolmogorov [1968], Li and Vitányi [1997]). However, this result only holds for infinitely long sequences, and therefore it cannot be applied to finite (sometimes short) genomes.

If we are only interested in the entropy evolution of the genome, and not in the particular value estimation, we can resort to the following trick: the genetic algorithm can be modelled as a thermodynamic system which evolves over time, where every population is just a measurement by an observer, i.e. the system is modeled as a statistical ensemble of genomes and each measurement is just a sample from that ensemble.

Now we can measure the entropy evolution of the system from two different viewpoints. The first is the system itself, where the entropy is calculated a la Shannon (equation 2) by estimating the probabilities of the loci alleles, using the frequencies of the ensemble sample.

The second way of measuring the entropy is from the viewpoint of the observer, where a measurement is made of the population at each time step, and the information about the system is updated, i.e., the observer measures the system at time $t$ and stores this information $S_t$. At time $t + 1$ the observer makes another measurement and substitutes $S_t$ by $S_{t+1}$. The entropy variation due to this substitution can be calculated for both equilibrium and non-equilibrium thermodynamic systems [Zurek, 1989a,b, Bennett, 1982]. Since evolution cannot be modeled as a system in equilibrium, the second case applies: the mutation and generational replacement operators may increase or decrease the entropy of the system [Vose, 1999, Wright, 2005].

Thus the entropy variation from the observer viewpoint is $K(S_t)-K(S_{t+1})$ bits. As $K(\cdot)$ is an incomputable measure, we estimate it by using the Lempel-Ziv algorithm [Ziv and Lempel, 1978] as $\lim_{n\to\infty}(1/n)|LZ(x)| = (1/n)K(x|n)$, where $x$ is an infinite string and $|LZ(x)|$ is the size of the same string compressed [Cover and Thomas, 1991]. Now our measurement of the system at time $t$, $|S_t|$ is much larger than with equation 3 (just the genome), so the estimation becomes possible.

## EXPERIMENTS

We want to test whether the evolution of K (population sample) can help in the study of the evolution of H (genome). Both measurements have their limitations, but their agreement would provide evidence that the entropy evolution is being studied correctly.

We have evaluated this experimentally, using the genetic algorithm proposed by Hayashi et al. [2007], which is able to reproduce sexual behaviour in a vary detailed way, because it includes several features absent in the Adami et al. [2000] experiments, such as sexual reproduction, different inter-locus and intra-locus interactions across the genotypic or phenotipyc distance, and the evolutionary mechanisms of mutation and natural selection.

We have implemented and run the same simulation proposed by Hayashi et al. The model's sexual dynamics can be summarized as follows: there are two different sexes (male and female). The likelihood that a female with trait a will mate with a male with trait b is defined by $\psi(d) = e^{-\alpha d^2}$, where d is the genetic or phenotypic distance measuring compatibility between the sexes, and $\alpha$ is a parameter that represents the compatibility between traits. The value of $\alpha$ used in the simulations is 0.005. The overall number of offspring produced by a female in each coupling is given by

$$W_f = B_{\max} e^{-sc(P-P_{opt})^2}$$

where P is the proportion of males with which the female has been able to mate. The parameter $P_{opt}$ (which can take the values {0.2, 0.4, 0.6, 0.8}) defines the fertility of the female. The parameter sc (ranging from 1.02 to 4×1.02) stands for sexual conflict selection in females. $B_{max}$ is the maximum possible number of offspring (5). The sex and the father are randomly chosen for each offspring, and the number of males each female encounters is n = 20.

Twenty two different experiments have been performed. K (population sample) and H (genome) have been estimated for each generation step with the methods described above. The system evolved for 10,000 generations with a population size of 1,000 individual genomes.

In all our results, both measures were highly correlated (see fig. 1), giving evidence that the dual measurement of the evolution of entropy by means of Kolmogorov complexity confirms the use of the Shannon entropy.

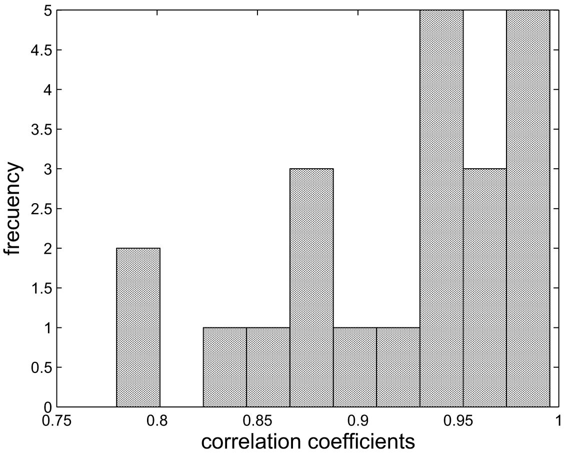


Figure 1: Histogram of the 22 experimental correlation coefficients.

## DISCUSSION

When sexual dynamics are introduced in the system, the increase of complexity observed by Adami et al. [2000] is not present anymore.

The typical result observed in our experiments is a chaotic behavior of the entropy (fig. 2). Only large genome lengths or high female mating rates ($P_{opt}$) escape from this (the other parameters seem to have little importance). The effect of this is to increase the autocorrelation of the entropy time series and provoke the rise of a few entropy bumps during a small number of generations (figs. 3 and 4).

Our hypothesis for this behavior of the entropy is the absence of natural selection in the Hayashi et al. [2007] model, which could explain the similarities between male and female evolution (fig. 5). Without natural selection, the environment for females is reduced to random boundary conditions (mutations). On the other hand, males are selected by females as mating partners. In this way, females can be considered to become the environment for males, since they determine the way in which the entropy of the males evolves. On the other hand, females have no environment to adapt to. Perhaps if the pressure of natural selection was applied both to male and female (not necessarily in the same way), more complex patterns in the behavior of their entropies would appear. The fact that natural selection is not taken into account may be the cause of the differences in entropy evolution between the models by Adami et al. [2000] and Hayashi et al. [2007], and the reason why global decreases in entropy are not observed in the latter.

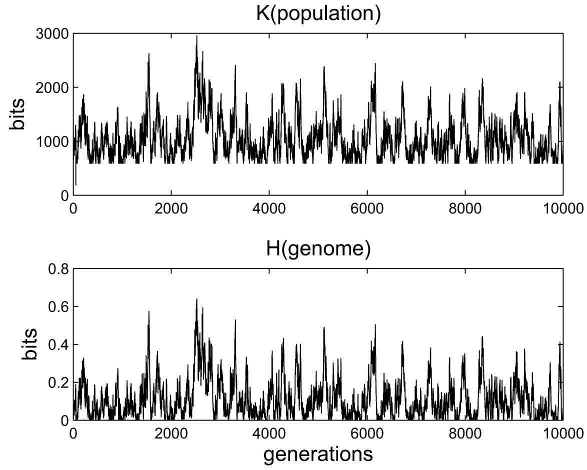


Figure 2: Example of *continuous evolutionary chase* without *genetic differentiation* obtained with parameters $B_{max} = 5$, $P_{opt} = 0.2$, $\mu = 5.0 \times 10^{-5}$, $\alpha$=0.05, sc = 1.02, 2 loci, phenotypic model: additive. Experimental correlation coefficient: 0.9956.

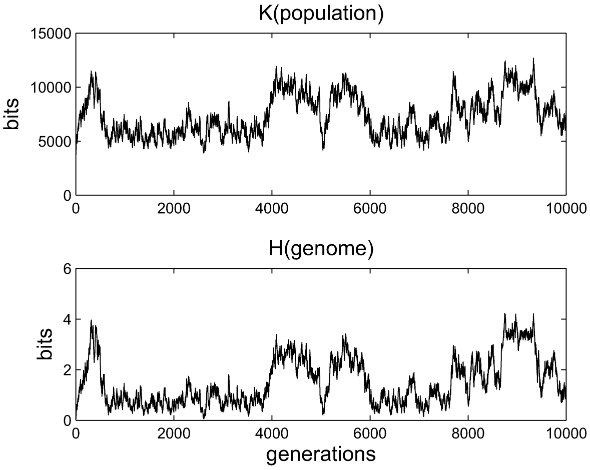


Figure 3: Example of *differentiation* without *speciation*, obtained with parameters $B_{max} = 5$, $P_{opt} = 0.8$, $\mu = 5.0 \times 10^{-5}$, $\alpha = 0.05$, sc = 1.02, 8 loci, phenotypic model: co-dominance. Experimental correlation coefficient: 0.9741.

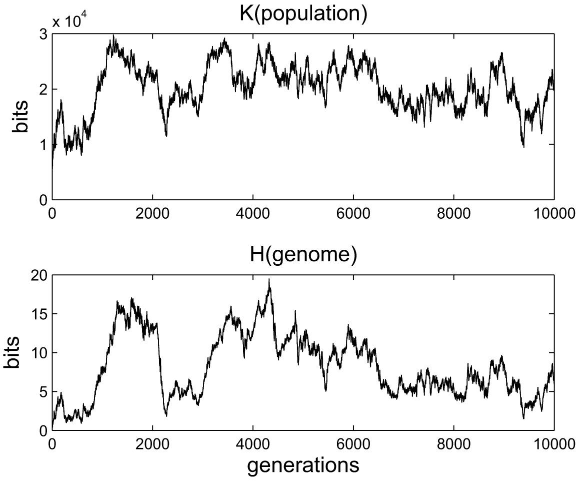


Figure 4: Example of genetic differentiation without co-evolutionary chase or simpatryc speciation, obtained with parameters $B_{max} = 5$, $P_{opt} = 0.2$, $\mu = 5.0 \times 10^{-5}$, $\alpha = 0.05$, sc = 1.02, 32 loci, phenotypic model: dominance. Experimental correlation coefficient: 0.8571.

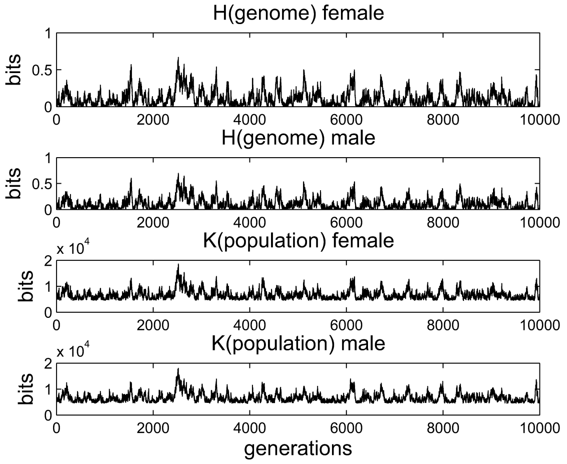


Figure 5: Same parameters as in fig. 2, with entropy calculation decomposed by sex.

## CONCLUSIONS AND FUTURE WORK

Studying a genetic algorithm from the observer point of view allows us to have large-scale estimates of the entropy evolution via Kolmogorov Complexity. This overcomes many limitations that arise from epistatic effects between loci and provide an easy way to study the dynamics of the algorithm without resorting to complex mathematical trickeries.

We have also used this methodology to study the effect of sexual reproduction in terms of the evolution of complexity as a decrease in entropy. We show that, when sexual reproduction is present, the population enters a chaotic regime of complexity driven by the complexity drifts of the female organisms. We suggest that this might happen because in our experiments female organisms evolve chaotically, without natural selection, while male organisms evolve using females as boundary conditions, which gives rise to an overall chaotic evolution of complexity.

In next immediate step we plan to introduce natural selection in the experiments and find whether this will change the evolution of complexity. We plan to do it by implementing the typical natural selection operators from genetic algorithms, such as tournament selection, steady state-selection, and so forth. We would also analyze the effects of different types of selection pressure on the females, such as food availability and other environmental aspects which up to now have not been considered. We conjecture that introducing natural selection will remove the chaotic complexity dynamics and might probably get closer to what Hayasi et. al. formerly reported: an increase in complexity attained by removing those genetic mutations that do not improve the fitness function.

The reason why we have not taken those aspects into account in our current implementation is the fact that we are not addressing any flavor of genetic algorithms, but studying the general mechanics of this family of algorithms. In the future we may try to address this point too.

## ACKNOWLEDGEMENTS

This work has been partially sponsored by MICINN, project TIN2008-02081/TIN and by DGUI CAM/UAM, project CCG08-UAM/TIC-4425.

We would like to thank Carlos Castañeda for his help in the implementation and simulations.

## REFERENCES

C. Adami and N. J. Cerf. Physical complexity of symbolic sequences. *Phys. D*, 137(1-2):62–69, 2000.

C. Adami, C. Ofria, and T.C. Collier. Special feature: Evolution of biological complexity. *Proceedings of the National Academy of Sciences*, 97(9):4463–4468, 2000.

G.P. Basharin. On a Statistical Estimate for the Entropy of a Sequence of Independent Random Variables. *Theory of Probability and its Applications*, 4:333, 1959.

C.H. Bennett. The thermodynamics of computation. *International Journal of Theoretical Physics*, 21(12):905–940, 1982.

T.M. Cover and J.A. Thomas. *Elements of information theory*. Wiley New York, 1991.

T.I. Hayashi, M.D. Vose, and S. Gavrilets. Genetic differentiation by sexual conflict. *Evolution*, 61:516–529(14), March 2007.

H. Herzel, W. Ebeling, and A.O. Schmitt. Entropies of biosequences: The role of repeats. *Physical Review E*, 50(6):5061–5071, 1994.

A.N. Kolmogorov. Three approaches to the quantitative definition of information. *International Journal of Computer Mathematics*, 2(1):157–168, 1968.

M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 1997.

M.D. Vose. *The Simple Genetic Algorithm: Foundations and Theory*. MIT Press, 1999.

A.H. Wright. *Foundations of genetic algorithms*. Springer, 2005.

J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *Information Theory, IEEE Transactions on*, 24(5):530–536, Sept. 1978.

W. H. Zurek. Thermodynamic cost of computation, algorithmic complexity and the information metric. *Nature*, 341:119–124, Sept. 1989a.

W.H. Zurek. Algorithmic randomness and physical entropy. *Physical Review A*, 40(8):4731–4751, 1989b.

## SUMMARY OF AUTHOR BIOGRAPHICAL DATA

Manuel Cebrián is a doctor in Electrical Engineering and Computer Science, formerly at Brown University, then in Telefonica Research, currently at the Human Dynamics Lab (The Media Lab), Massachusetts Institute of Technology.

Manuel Alfonseca is a doctor in electronics engineering and computer scientist, formerly a Senior Technical Staff Member of IBM, currently a full professor at the Universidad Autónoma of Madrid.

Alfonso Ortega is a doctor in computer science, formerly in LAB2000, currently a professor at the Universidad Autónoma of Madrid.

# ADAPTING AN EVOLUTIONARY ALGORITHM WITH EMBEDDED SIMULATION AND PSEUDO-RANDOM NUMBER GENERATION FOR THE CELL BROADBAND ENGINE

Sofie Van Volsem, Sven Neirynck
Department of Industrial Management
Ghent University
Technologiepark 903
BE-9052 Zwijnaarde, Belgium
e-mail: {Sofie.VanVolsem, Sven.Neirynck}@UGent.be

**KEYWORDS**
PlayStation®3, cell processor, SIMD, normal random numbers, simulation, evolutionary algorithm

## ABSTRACT

For the problem of optimizing inspection strategies in multi-stage production systems, a metaheuristic consisting of an evolutionary algorithm with embedded simulation was developed in Van Volsem et al. (2007), Van Volsem (2009) and Van Volsem (accepted for publication, 2009). The metaheuristic requires normally distributed pseudo-random numbers; the time needed for this random number generation is a substantial fraction of the total computation time. In an effort to reduce the computation time, the metaheuristic was adapted for computation on the Cell Broadband Engine. The proposed adaptation is twofold: we propose a way to make the metaheuristic suitable for fast multicore computation, and secondly, the potential of SIMD computation for speeding up the random number generation process and the metaheuristic is investigated.

## INTRODUCTION

Traditional computer software is written for serial computation. To solve an optimization problem, an algorithm or metaheuristic is constructed and implemented as a serial stream of instructions. These instructions are executed on a central processing unit (CPU) on one computer.

Parallel computing uses multiple processing elements simultaneously to solve a problem. This is accomplished by breaking the problem into independent parts so that each processing element can execute its part of the algorithm simultaneously with the others. The processing elements can be diverse and include resources such as a single computer with multiple processors, several networked computers, specialized hardware, or any combination of the above.

Today most commodity CPU designs include single instructions for some vector processing on multiple (vectorized) data sets, typically known as SIMD (Single Instruction, Multiple Data). Modern video game consoles and consumer computer-graphics hardware rely heavily on vector processing in their architecture. In 2000, IBM, Toshiba and Sony collaborated to create the Cell Broadband Engine (Cell BE), consisting of one traditional microprocessor (called the Power Processing Element or PPE) and eight SIMD co-processing units, or the so-called Synergistic Processor Elements (SPEs), which found use in the Sony PlayStation®3 among other applications.

The computational power of the Cell BE or PlayStation®3 can also be used for scientific computing. Examples and applications have been reported in e.g. Kurzak et al. (2008), Bader et al. (2008), Olivier et al. (2007), Petrini et al. (2007).

In this paper, the potential of using the PlayStation®3 for speeding up metaheuristic optimization is investigated. More specifically, we propose an adaptation of an evolutionary algorithm with embedded simulation for inspection optimization. Thereto, two issues need to be addressed:

- the metaheuristic itself needs to be adapted to make it suitable for parallel computation and vector processing, and
- the random number generation required by the metaheuristic has to be adapted for parallel computing as well.

The main mechanism of achieving this goal is through parallelization and vectorization. The main obstacles in the way of parallel execution are data hazards, which prevent simultaneous execution of instructions with dependencies between their arguments, and control hazards, which result from branches and other instructions changing the Program Counter (Kurzak et al. 2008). In other words:

- Instructions can be grouped together only if there is no data dependency between them.

- Communication and synchronization between the different subtasks are typically one of the greatest obstacles to getting good parallel program performance.

Moreover, the potential speed-up of an algorithm on a parallel computing platform is limited by Amdahl's law, which states that a small portion of the program which cannot be parallelized will limit the overall speed-up available from parallelization.
The remainder of the paper is organised as follows: in Section 2, the original problem and metaheuristic solution approach is described. Section 3 reports on the adaptation of the metaheuristic, while the potential of SIMD computation for speeding up the random number generation process is investigated in Section 4. Section 5 summarizes and concludes.

## INSPECTION OPTIMIZATION WITH AN EVOLUTIONARY ALGORITHM AND SIMULATION

Efficient production quality control is a major issue to manufacturers. Efficient economic inspection strategies ensure the required output quality while minimizing the total inspection cost. Generally speaking, more and tighter inspection will induce a higher product quality – in terms of meeting product specifications– but will also result in higher costs of inspection, scrap and rework. An economic inspection plan will balance this trade-off.
Consider a serial multistage production system (MSPS) in which products travel sequentially from stage 1 to stage $n$ and inspection of products is performed by $k$ ($k \leq n$) inspection stations. At each stage, a manufacturing action is performed on or with the products, before moving on to an inspection station, or to the processing station of the next stage in case of no inspection.
An inspection strategy for an MSPS decides on:

1. the number and location of inspection stations;
2. the rigor of the inspections (inspection limits) for each inspection station.
3. the number of inspections executed (sample size or sampling frequency and acceptance number) for each inspection station;

The problem facing the MSPS inspection planner thus consists of finding the combination of these inspection parameters in order to minimize the total expected inspection cost ($TIC$). This is a complex joint optimization problem; addressed in Van Volsem et al. (2007), Van Volsem (2009) and Van Volsem (accepted for publication, 2009).
Their joint optimization method consists of embedding Monte Carlo simulation (to compute the serial $n$-stage MSPS subject to inspection) in an Evolutionary Algorithm (EA) (to perform the actual optimization). For each of the $n$ stages, the EA proposes the inspection type ($F$ for full inspection, $S$ for sampling inspection, $N$ for no inspection), the upper and lower inspection limits ($UIL$ and $LIL$), and the acceptance sampling parameters ($s$ and $t$) where appropriate.
The following notation is used:

| | | |
|---|---|---|
| $K$ | = | batchsize |
| $n$ | = | number of process stages |
| $p_i'$ | = | fault occurrence in stage $i$ |
| $LIL_i$ | = | lower inspection limit in stage $i$ (variable) |
| $UIL_i$ | = | upper inspection limit in stage $i$ (variable) |
| $LS_n$ | = | lower specification limit after stage $n$ (fixed) |
| $US_n$ | = | upper specification limit after stage $n$ (fixed) |
| $s_i$ | = | sample size for stage $i$ |
| $t_i$ | = | acceptance number for stage $i$ |
| $d_i$ | = | number of bad items after stage $i$ |
| $c_{T,i}$ | = | unit test cost in stage $i$ |
| $c_{R,i}$ | = | unit rework cost in stage $i$ |
| $c_P$ | = | unit penalty cost (after stage $n$) |
| $TC_i$ | = | test cost in stage $i$ |
| $RC_i$ | = | rework cost in stage $i$ |
| $\alpha_{F,i}$ | = | 1 if $F$ is selected in stage $i$, = 0 otherwise |
| $\alpha_{S,i}$ | = | 1 if $S$ is selected in stage $i$, = 0 otherwise |
| $TTC$ | = | total test cost |
| $TRC$ | = | total rework cost |
| $TPC$ | = | total penalty cost |
| $TIC$ | = | total inspection cost |

The $TIC$ is calculated as follows:

$$TIC = TTC + TRC + TPC \tag{1}$$

with

$$TTC = \sum_{i=1}^{n} TC_i \tag{2}$$

$$TRC = \sum_{i=1}^{n} RC_i \tag{3}$$

$$TPC = c_P.d_n \tag{4}$$

and with

$$TC_i = c_{T,i}.(\alpha_{F,i}.K + \alpha_{S,i}.s_i) \tag{5}$$

$$RC_i = c_{R,i}.p_i'.\alpha_{F,i}.K \tag{6}$$

Determining the optimal inspection strategy, i.e. the whole of inspection decisions that minimize the $TIC$, requires the determination of inspection options $\alpha_i$ and the corresponding inspection limits ($LIL_i$, $UIL_i$) and sampling parameters ($s_i$, $t_i$), for all stages $i = 1, ..., n$.
This is what the proposed Evolutionary Algorithm does. Evolutionary Algorithms are adaptive heuristic search methods mimicking selective breeding, where offspring

are sought which have certain desirable characteristics, determined at the genetic level by combination of the parents' chromosomes. In a similar way, in seeking better solutions, EA's combine pieces of existing solutions: new generations of offspring are generated through an iteration process until a convergence criterion is met. The basic concepts were developed by Holland (1975) and were forged into a problem solving methodology for complex optimization problems by De Jong (1975) and Goldberg (1989).

There are four main parts in the EA paradigm, namely the problem representation and initiation, the objective function evaluation (fitness calculation), the parent selection, and the actual evolutionary reproduction of candidate solutions.

### Problem representation and initiation

Every proposed solution is represented by a vector of the independent variables (inspection decision variables), coded as a chromosome constituted by as many *genes* as the number of independent variables. The chromosomes used in the EA we propose, consist of a set of "character" values ($F$, $N$ or $S$), real values ($LIL_i$ and $UIL_i$) and integer values $(s_i, t_i)$.

We used a population size $M$ of 50 initial solutions. From this pool, some are selected (parents) to construct new solutions (children). The construction algorithm for the initial population consists in randomizing the characters $(N, S, F)$, and randomizing the limits by allowing (symmetrical) variation from the original limits by a certain user defined percentage.

### Objective function evaluation (fitness calculation)

For every candidate solution its fitness as a possible parent has to be evaluated, where fitness refers to measure of profit or goodness to be maximized while exploring the solution space. We use a straightforward normalization procedure to calculate the fitness value $f$ for each solution $j$ in a population of $M$ solutions:

$$f_j = \frac{1/\,TIC_j}{\sum_{k=1}^{M}(1/TIC_k)} \tag{7}$$

### Parent selection

Parent selection for producing offspring is done as in Holland's original Genetic Algorithm, i.e. for each reproduction two parents are chosen: one parent is selected on its fitness basis, the other is chosen randomly. The idea behind this is that the parent chosen for its fitness ensures genetic quality, while the random parent ensures genetic diversity.

### Reproduction

In our algorithm, the new generation consists of $(M-1)$ children, the $M^{th}$ solution in the next generation population is the best solution from the previous generation (=*elitism of 1*). Generating offspring is performed in two steps: first crossover is applied, then the inspection limits are adapted. After these two steps, reproduction is completed and the children thus obtained can populate the new generation. This way, the simultaneous determination of inspection parameters is achieved.

Our crossover operator randomly selects a crossover point, and constructs two new solutions by exchanging the tails of both parents. Instead of mutation, inversion is used (see Reeves (1993; pg. 173)).

**Algorithm 1** original EA

```
Create initial sorted population
for generation = 1 to number of generations do
  Create offspring
  for solution = 1 to $M$ do
    for stage = 1 to $n$ do
      Calculate process values
      Calculate inspection cost
    end for
    Calculate TIC
  end for
  Sort population
end for
Take winner
```

## ADAPTATION OF THE EA

Kurzak et al. (2008) state that while not all problems SIMDize well, most can benefit from it one way or another.

To adapt the above metaheuristic for computation on the Cell Broadband Engine, we need to parallelize it to make it suitable for computation on the SPEs, and consequently the code has to be vectorized for in order to make efficient use of the SPEs.

First we have to choose which parts will run on the SPEs and what on the PPE. Next step is to SIMDize the SPE code. Caclulating the TICS of an inspection strategy requires a fair amount of computational power and has no data dependencies. This is therefore an ideal candidate to run on the SPE's. The creation of offspring for new generations will be done on the central PPE, which will then communicate the inspection strategies to the SPEs who will calculate the TICs in parallel.

The TIC calculation on the SPE's consists of calculating the process values and subject them to the selected inspection strategy. Each solution is simulated 50 times, the average TIC is returned. This simulation requires a lot of normal random numbers, how to adapt the random number generation for parallel processing is the

**Algorithm 2** adapted version of the EA

```
Create initial sorted population
for generation = 1 to number of generations do
  for solution = 1 to $M$ do
    Create offspring
    Calculate TIC on SPE (IN PARALLEL)
  end for
  Sort population
end for
Take winner
```

subject of the next section.
How is the TIC calculation now implemented on an SPE? Keeping in mind that the memory of an SPE is limited to 256k, we have decided to calculate it one process value at a time, through all $n$ stages instead of complete batches stage by stage. This is feasible as the processing of the values is not interdependent. This way we needn't worry about the 256k memory which would otherwise become a problem if the batchsize becomes to big.
How do we SIMDize the inspection? Normally this code consists of a lot of branches:

```
Do we need to inspect? YES/NO
Is the value between inspection limits? Y/N
Are we using sampling inspection? Y/N
Do we need to switch to full inspection? Y/N
```

In SIMD we cannot have the conditional branches as the same code needs to run on all the elements in the vector. We implemented a standard technique of processing the instructions for both branches of the conditional branch and at the end select the right value with the instruction `SPUsel`.

## NORMAL PSEUDO-RANDOM NUMBERS

In many simulation and Monte Carlo programs, a substantial fraction of the computation time is used in generating pseudo-random numbers (Brent 1998). Vector or parallel computation can significantly contribute in accelerating the simulation process. However, parallel computation for Monte Carlo programs in itself also brings about some difficulties that cannot be overlooked:

- The requirements for parallel random number generators (RNGs) are more stringent than those for sequential RNGs. If a simulation is to be run on a multi-processor machine, it is of the essence to ensure that the random numbers used by each processor are independent, or equivalently, to ensure that the sequences of random numbers used by each processor are disjoint.

Different applications require pseudo-random numbers with different distributions (uniform, normal, exponential, Poisson, etc.). The algorithms used to generate these random numbers usually rely on a good source of uniform random numbers, which are then transformed to random numbers with other distributions. The algorithms used can be divided in the following groups:

**Inversion methods** are based on the observation that continuous cumulative distribution functions (cdfs) range uniformly over the interval $[0, 1]$. If $u$ is a uniform random number on $[0, 1]$, then a random number $X$ from a continuous distribution with specified cdf $F$ is obtained using $X = F^{-1}(U)$. Subject to the restriction that the distribution is continuous, this method is generally applicable (and can be computationally efficient if the cdf can be analytically inverted)

**Transformation methods** provide an alternative in cases where cdf inversion too computationally expensive in practice for some probability distributions. The Box-Müller transform is an example of such an algorithm. It produces two normally distributed random numbers from a pair of uniformly distributed random numbers. If $u1$ and $u2$ are independent random variables that are uniformly distributed on [0, 1], then

$$\begin{aligned} z_0 &= R\cos(\Theta) = \sqrt{-2\ln u_1}\cos(2\pi u_2) \\ z_1 &= R\sin(\Theta) = \sqrt{-2\ln u_1}\sin(2\pi u_2) \end{aligned}$$

are independent random variables with a standard normal distribution.

**Acceptance-rejection methods** also provide an alternative in cases where the functional form of the required distributions makes it difficult or time-consuming to generate random numbers using inversion methods. As the previous two methods, acceptance-rejection methods require uniform random numbers. In this method it is assumed that the probability distribution $F$ we wish to simulate has a pdf $f(x)$. The basic idea is to find an alternative probability distribution $G$, with pdf $g(x)$, from which we already have an efficient algorithm for generating from, but also such that the function $g(x)$ is "close" to $f(x)$. In particular, we assume that the ratio $f(x)/g(x)$ is bounded by a constant $c > 0$. The algorithm for generating $X$ distributed as $F$ proceeds as follows:

1. Choose a pdf $g$.
2. Find a constant $c$ such that $f(x)/g(x) \leq c \,; \forall\, x$
3. Generate a uniform random number $u$
4. Generate a random number $v$ from $g$
5. If $c * u \leq f(v)/g(v)$ , accept and return $v$. Otherwise, reject $v$ and go to step 3

For efficiency, a "cheap" method is required for generating random numbers from $g$, and the scalar $c$ should be small.

These methods all either involve the computation of mathematical functions such as sines, cosines and logarithms, which are slow in comparison to the time required to generate a uniform random number, or require on average more than one uniform random number for each normal random number. From this it evidently follows that normal RNGs based on transforming uniform random numbers are slower than uniform RNGs. Leva (1992) compared several of the best acceptance-rejection methods an found that they are at least five times slower than a fast uniform RNG on the same machine.

Brent (1998) argues that the most well-known and widely used methods for normal RNG often do not vectorize well. He therefore suggests vectorized implementations of the "old-fashioned" Box-Müller transformation.

We follow this suggestion; the RNG used to generate the uniform random numbers is the SFMT (SIMD-oriented Fast Mersenne Twister, Saito and Matsumoto (2006)), followed by our own implementation of the Box-Müller transform. Standard mathematical functions such as sines and cosines are calculated using the Universal SIMD Mathlibrary (2009) `libsimdmath`.

## SUMMARY AND CONCLUSIONS

We design an optimized parallel implementation of an evolutionary algorithm and simulation for optimizing inspection strategies for multi-stage processes on the PlayStation®3. We adapted the algorithm to eliminate branches and optimized the code using standard techniques such as loop unrolling and vectorization. We adapted the random number generation process: we developed and implemented a Box-Müller transform on uniform random numbers generated with an 128-bit Mersenne twister. The original algorithm calculation time was >1 hour for 200 generations; re-writing the code with SIMDizing the RNG led to a calculation time of 5'59" on a single core of a 2.5GHz AMD Phenom processor. The further porting and optimizing of the code to make it suitable for running on the cell broadband engine led to a calculation time of 14" on a PlayStation®3. We thus realized a speedup factor of more then 256 in comparison with the original algorithm in Van Volsem et al. (2007), and showed the PlayStation®3 suitable for scientific computing.

## REFERENCES

Bader D.; Chandramowlishwaran A.; and Agarwal V., 2008. *On the design of fast pseudo-random number generators for the cell broadband engine and an application to risk analysis. IEEE Transactions on the 37th International Conference on Parallel Processing.*

Brent R., 1998. *Random number generation and simulations on vector and parallel computers. Proceedings of the 4th International Euro-Par Conference*, 1–20.

De Jong K.A., 1975. *An Analysis of the Behaviour of a Class of Genetic Adaptive Systems.* Ph.D. thesis, University of Michigan Press.

Goldberg D., 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison Wesley, NY.

Holland J.H., 1975. *Adaptation in Natural and Artificial Systems.* University of Michigan Press.

Kurzak J.; Buttari A.; Luszczek P.; and Dongarra J., 2008. *The PlayStation3 for high performance scientific computing. Computing in Science and Engineering*, 10, no. 3, 84–87.

Leva J., 1992. *A fast normal random number generator. ACM Transactions on Mathematical Software*, 18, 449–453.

Olivier S.; Prins J.; Derby J.; and Vu K., 2007. *Porting the GROMACS Molecular Dynamics Code to the Cell Processor. Proceedings of the 8th IEEE International Workshop on Parallel and Distributed Scientific and Engineering Computing.*

Petrini F.; Fossum G.; Fernandez J.; Varbanescu A.L.; Kistler M.; and Perrone M., 2007. *Multicore surprises: lessons learned from optimizing Sweep3D on the cell broadband engine. IEEE Transactions on the 2007 International Parallel and Distributed Processing Symposium.*

Reeves C.R., 1993. *Modern Heuristic Techniques for Combinatorial Problems.* Blackwell Scientific Publications.

Saito M. and Matsumoto M., 2006. *SIMD-oriented Fast Mersenne Twister: a 128-bit Pseudorandom Number Generator. Proceedings of the the 7th International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing.*

Universal SIMD Mathlibrary, 2009. URL `http://webuser.fh-furtwangen.de/~dersch/`.

Van Volsem S., 2009. *Joint optimization of all inspection parameters for multi-stage processes: algorithm, simulation and test set. Proceedings of the 16th European Concurrent Engineering Conference.*

Van Volsem S., accepted for publication, 2009. *Joint optimization of all inspection parameters for multi-stage processes: evolutionary algorithm and simulation. International Journal of Innovative Computing and Applications.*

Van Volsem S.; Dullaert W.; and Van Landeghem H., 2007. *An Evolutionary Algorithm and Discrete Event Simulation for Optimizing Inspection Strategies for Multi-Stage Processes. European Journal of Operational Research*, 179, 621–633.

## AUTHOR BIOGRAPHY

**SOFIE VAN VOLSEM** received a MSc degree in Chemical Engineering from Ghent University in 1998 and a PhD in Engineering Sciences from the same institution in 2006. She worked in industry as a process & quality engineer before returning to academia. After being with the University of Antwerp for 6 years and the University College of West-Flanders for nearly 2 years, she currently holds a post-doc position at Ghent University. She teaches Quality & Industrial Statistics. Her research interests are quality management, quality and reliability issues in supply chains, management applications of metaheuristics.

**SVEN NEIRYNCK** graduated as MSc in Computer Sciences at Ghent University in 1997. After 10 years of working as a systems engineering manager, he currently divides his time between IT consultancy with expertise in storage, archiving, data security and HPC systems; and research in the area of simulation at Ghent University.

# BEHAVIOURAL MODELLING

# MODELLING MOTORWAY DRIVING BEHAVIOUR USING MICROSCOPIC VEHICLE TRAJECTORY DATA

Abs Dumbuya[1] and George Lunt [2]
[1]TRL, Crowthorne House, Nine Mile Ride, Wokingham, Berkshire, RG40 3GA, UK,
[2]AECOM, Portwall Place, Bristol, BS1 6NB, UK
E-mail: adumbuya@trl.co.uk

## KEYWORDS

Approximation techniques, Behavioural science, Data enrichment, Model design, Validation

## ABSTRACT

The paper presents a micro-simulation model of motorway driving using vehicle trajectory data to define key behaviours such as longitudinal and lateral movements. Micro-simulation plays an important role in transport research. In particular it can be used to appraise new transport schemes, help predict short/medium term traffic conditions in real time, and provide a realistic driver behaviour model for implementation within a driving simulator. In the past microscopic model developers have tended to build models 'top-down' due to the cost of acquiring suitable microscopic field data. A top-down approach has a principal aim of creating microscopic models that represent aggregated macroscopic field data correctly, with less attention to the behaviour at the level of individual drivers. The aim of the project described in this paper was to build a microscopic model from the 'bottom-up'. A bottom-up approach uses microscopic field data (namely vehicle trajectory data) to build and calibrate microscopic models, and has a principal aim to replicate the behaviour of individual drivers. It is anticipated that if the microscopic behaviour is modelled with sufficient accuracy then the aggregated macroscopic statistics (e.g. journey times, capacities, queue lengths, etc.) will also be correct, and greater confidence can be given to the model predictions.

## INTRODUCTION

### Vehicle trajectory data

The research project makes use of a novel vehicle trajectory dataset - Individual Vehicle Data (IVD) (see Lunt 2005; Lunt and Wilson 2003) that is owned by the Transport Research Laboratory (TRL). A detailed description of the micro-simulation framework and how the data was created and used in a Neural Network behavioural model can be found in (Dumbuya et al, 2006; Dumbuya et al, 2007). The work was further developed in 2007/08 to enhance the framework to model two key behaviours, namely longitudinal and lateral driver behaviour in free-flow motorway conditions (see discussion on the behavioural theory of traffic dynamics for homogenous multi-lane freeways, Daganzo 1999; Gunay, 2007; Laval and Leclercq, 2007; Taylor et al, 2008). Vehicle trajectory data is an appropriate data source to build microscopic models from the bottom-up. This data describes the path of individual vehicles over time, giving a time series of individual speeds, accelerations, and lateral movements. When combined with surrounding vehicle trajectories it is possible to build an understanding of how drivers interact with each other, and allow appropriate models to be built. Although trajectory data alone does not provide information on a particular driver's thought processes, it still provides a valuable resource to help understand vehicle interactions at a microscopic level.

The use of vehicle trajectory data to build microscopic models is not a completely new concept. Recently the capture of video footage and the subsequent automatic/manual re-identification of vehicles has provided model developers with trajectory information (see Hoogendoorn et al, 2005). Survey vehicles have also been used and can provide very frequent time-series trajectory data from one vehicle. However, due to the manual effort required to track vehicle trajectories from video footage, the data sets typically only span ½ to 1 hour. Survey vehicle data is also expensive to collect and so only a few drivers are monitored in a trial; consequently the sample will not represent the full variation in driver behaviour. It is also possible that drivers of survey vehicles might behave differently knowing that they are being monitored and so provide a biased sample. For both reasons the data sets do not provide a sufficiently large, representative sample of the driving population.

Individual vehicles recorded at consecutive MIDAS[1] inductance loop sites were automatically re-identified (see Lunt,2005for a description of the method) across a range of 500 metres at fixed 100metre intervals, proving trajectory information.90,000vehicle trajectories were recorded over a consecutive period of 3 days (8th to 9th December 2003). The data was recorded at Junction 6 of the M42 near Birmingham UK, in-between the diverge and merge points, as shown in Figure 1. For each trajectory at each 100 metre site statistics were recorded detailing the measured vehicle speed, length, time headway to vehicle ahead, lane, and timestamp.

This unique data source is sampled at fixed locations. Video and probe vehicle trajectory data is sampled at fixed time intervals. The close proximity of the sites allows trajectory data of a sufficient sampling frequency to extract longitudinal and lateral driver behaviour in free-flow

[1] MIDAS (**M**otorway **I**ncident **D**etection and **A**utomatic **S**ignalling) is a Highways Agency ITS control system to provide roadside messages to drivers via Variable Message Signs (VMS)

conditions (vehicles travelling at 70mph are recorded at intervals of 3.2 seconds at fixed 100 metre intervals).

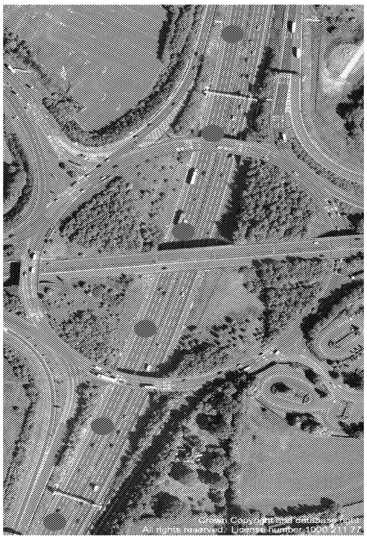

Crown Copyright and database right
All rights reserved. License number 1000 211 27

Figure 1: Locations of IVD Collection Sites M42 Junction 6 near Birmingham UK

If additional IVD were collected to cover high flow and low speed conditions then there would be uncertainty as to how well the existing, or modified, re-identification algorithms would correctly re-identify vehicles. Additionally the dataset would contain vehicle statistics at fixed distances, and in slow-moving conditions there could be long durations between successive trajectory measurements.

**Overview of longitudinal and lateral behaviour models**

Longitudinal models predict the longitudinal positions of vehicles. A major subset of these models, is concerned with *car following.* Car following is defined loosely as the interactions between two vehicles such that the following vehicle chooses to follow the lead vehicle, or, the condition is imposed such that the following vehicle is unable to overtake the lead vehicle and therefore slows down to follow the lead vehicle. A number of car following models may include:

- *Gazis-Herman-Rothery (GHR) models* (Chandler et al, 1958) - Based on a stimulus-response function i.e. the acceleration of a driver is proportional to the separation distance and relative speeds.
- *Linear (Helly) models* (Helly, 1959) - The acceleration of the current driver is proportional to whether the immediate driver in front (and two vehicles in front) was braking.
- *Safety distance or Collision Avoidance models (CA)* (Kometani and Sasaki, 1959 and Gipps, 1981) - Based on the concept of safe following distance manipulated by the Newtonian equation of motion i.e. what is the safe following distance to avoid a collision if the driver in front were to behave unpredictably?
- *Psycho-physical or Action Point models (AP)* (Michaels, 1963) - Based on the idea that a driver can operate in a number of driving regions. A threshold is used to determine the change from one region to the other. There are two seminal models proposed by (Fritzsche, 1994) and by (Wiedemann, 1974). The regions are characterised across two dimensions, the headway to the vehicle in front, and the closing (or opening) speed to the vehicle in front.

Lateral models (sometimes simplistically referred to as lane changing models) predict the sideways movements of vehicles, and are often simplified by assuming a vehicle's lateral location can only exist in a number of discrete states, or lanes. Although this is a simplification of the actual process, the models are still traditionally much more complex compared with longitudinal models. In lane changing, there are many considerations to be taken into account. The behaviour (or motives) of other drivers in the target lanes the current driver is intending to move to must be carefully assessed. Lane-changing opportunities can become available both in light uncongested and congested traffic conditions. In uncongested traffic, lane changing is considered feasible if there is a gap of sufficient size in the target lane so that the vehicle can move into the target lane safely, without forcing other vehicles in the target lane to slow down significantly. On the other hand, lane changing in congested traffic involves using 'forced' and 'co-operative' lane changing procedures, (Hidas, 2004, also see Daganzo 1999; Gunay 2007; Laval and Leclercq 2007; Taylor et al, 2008). A review of lane changing for the England Highways Agency model SISTM (SImulation of Strategies for Traffic on Motorways) identified three fundamental behavioural processes associated with lane changing, (Baguley et al, 2003), namely:

1. Lane change desire
2. Lane change availability
3. Lane change mechanics

A rule based approach is generally used to model lane changing. For example, in multi-lane roads a hierarchical set of rules is used to model lane changes (e.g. VISSIM,). In DRACULA, (Lin et al, 1995), the lane-changing model contains three steps: (1) obtain the lane-changing desires and define the type of changing, (2) select the target lane, and (3) change lane if all gaps are acceptable.

An inherent part of lane changing behaviour is gap acceptance. This is also a relatively complex behaviour that takes place when a driver wishes to insert themselves between vehicles in a different traffic stream. For example, in lane changing a driver must assess the critical gap length of the target lane to which he/she intends to move. Critical gaps are typically modelled as random variables to capture the variation in the behaviours of different drivers and for the same driver over time, (Ahmed et al, 1996). A gap is measured as headway between the most immediate pair of vehicles in the target traffic stream. As with car following, gap acceptance is also usually measured in time.

From the brief review of longitudinal and lateral models it is noted that the increased understanding of car following models, both from the traffic engineering and traffic psychology point of view, help to design better micro-simulation models. A previous limitation in psychological driver model development was the need for accurate and extensive behaviour data. However, now there are number of data sources to validate these models (see Brockfeld and Wagner 2006; Dougherty et al 2000; Bonsall, 2000).

Furthermore, a mixture of approaches including control (mechanistic/mathematical) and psychological (behavioural) would be required to build realistic models.

## MOTORWAY DRIVING BEHAVIOUR MODELS

### Development of Longitudinal Model

A psychological Action Point (AP) model was developed for the longitudinal (or car following) behaviour in motorway traffic. An AP phase diagram (in this case the boundaries are from the Fritzsche model) is shown in Figure 2. An example vehicle trajectory (thick black line) has been drawn on the diagram and the action points (occurrences where the trajectory passes into a new region and takes on new behaviour) are drawn as circles. An AP model contains behavioural regions defined between boundaries. When these boundaries are crossed then an action is taken to alter the existing behaviour. This 'existing behaviour' is typically a constant acceleration/deceleration, and the process mimics a driver fixing the accelerator pedal for a while, and then altering its position to another fixed point given a change in perceived conditions. The AP phase diagram operates in two dimensions for a typically lead-follower vehicle pair, defined over the space headway to the vehicle in front ($\Delta x$, vertical axis) and the difference in speed to the vehicle in front ($\Delta v$, horizontal axis). In this Fritzsche model, the driving regions are:

- Free driving, accelerate up to desired speed (green)
- Following, maintain the same speed (blue)
- Closing in, brake in order to reach the same speed as the vehicle in front (yellow), and
- Danger, brake sharply (red)

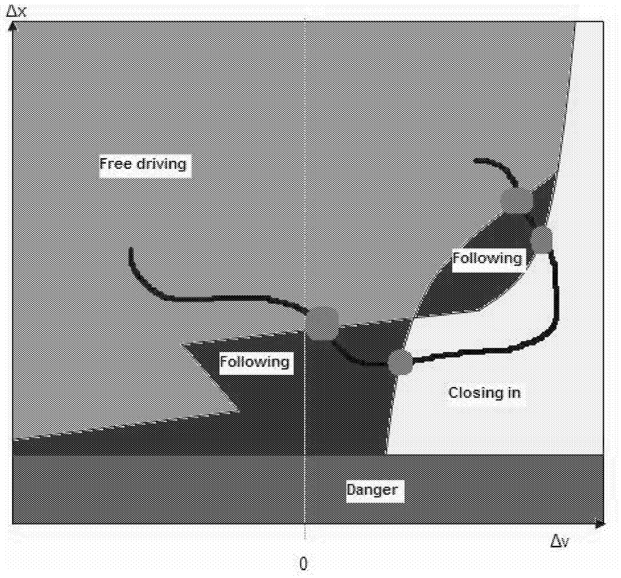


Figure 2: Illustration of Action Point Phase Diagram and Path of a Vehicle Trajectory (represented by the bold line)

Typically, with AP models, it is possible to cluster IVD to obtain behaviour in the different regions. The research project was concerned with extracting the most important factors that influence longitudinal behaviour, with an intention to consider more complex influences at a later time. For the purpose of this project, it was assumed that lane changing, and its affect on subject and surrounding vehicles, was a complex influence. Therefore, attempts were made to filter out trajectories affected by lane changes in order to facilitate the extraction of fundamental longitudinal driver behaviour from the trajectory dataset.

To consider vehicles unaffected by lane changes, filtering of the original 90,000 trajectories was conducted. This involved only considering vehicle trajectories that did not change lane and that had no lane change made immediately in front of them *across all the six inductance loop sites*. The detailed filtering process is described in (Lunt and Dumbuya, 2007).

Given the variation in driver behaviour in different lanes, and across vehicles of different lengths, a further filtering of vehicle trajectories that only travelled in lane 2, and those whose lengths were between 2 and 6 metres was made. This filtering reduced the dataset to a collection of vehicle trajectories that exhibited broadly similar characteristics.

In summary, the filtering process left vehicle trajectories with the following properties:

1. Trajectories that remained in lane 2 across all six sites
2. Trajectories unaffected by lane changes immediately in front of them across all six sites
3. Vehicles between 2 and 6 metres in length

The reduced dataset still consisted of 13,000 vehicle trajectories. This demonstrates one of the great benefits of the original IVD in that stringent filtering could take place and still leave a large number of vehicle trajectories (in this case 13,000) for model development. This kind of filtering is not feasible on much smaller datasets currently used for existing model development and calibration, since only a very small number of trajectories would remain.

Furthermore, only the trajectory statistics recorded at the third and fourth inductance loop site were considered. This was to reduce the chance of considering drivers that were not genuinely car following since there was no information on how vehicles behaved just before the data capture, and just after. By only considering statistics at the two middle sites, there was very high confidence that a lane change did not occur 200 metres before, and 200 metres after the measurements were made. The only possible errors would have been from incorrect re-identification within the original algorithm, or erroneous data from the loop sites. For a vehicle travelling at 70mph this corresponds to a time of 6.5 seconds before and after the statistics at the two central sites were measured.

New AP boundaries were constructed from the reduced vehicle trajectory dataset. By considering the speeds from loop 3 to loop 4 of each trajectory ($s_3$ and $s_4$), the acceleration over this distance (100 metres) is given by

$$\frac{s_4^2 - s_3^2}{100}$$

Calculating this acceleration for each trajectory, and subsequently categorising the trajectories into similar acceleration bands, gave rise to collections of vehicles that had similar responses. These responses were taken from the HA micro-simulation model SISTM - a microscopic simulation of traffic model which dynamically models ramp metering, speed control; takes into account shockwaves and exit management and allows the operation of those systems to be dynamically controlled in real time (see Baguley et al 2003 for details). The responses are defined as follows:

| | |
|---|---|
| Braking | less than -5 kph/s |
| Slowing down | between -5 and -0.5 kph/s |
| No Change | between -0.5 and 0.5 kph/s |
| Speeding Up | between 0.5 and 3.83 kph/s |
| Accelerating | greater than 3.83 kph/s |

The value -5 kph/s is a default value in SISTM and is defined as the braking rate adopted if the brake lights of the vehicle ahead are seen. The value 3.83 kph/s is a default value in SISTM and is defined as the maximum achievable acceleration rate when travelling at over 75% of desired speed. The values of -0.5 and 0.5 kph/s were manually chosen as a suitably insignificant acceleration and deceleration rates and within a driver's natural variation in maintaining a fixed speed.

### Development of Lateral Model

Lateral or lane changing models can be complex. A three stage lateral or lane changing model was developed for the project. This considers

1. Lane change desire
2. Lane change availability
3. Lane change mechanics

The justification for choosing this three-phase lane changing model was that the same approach was adopted in the HA SISTM tool developed by TRL, so was familiar to the developers. Also, although there is currently little field data to support model development for processes 1 and 3 the IVD trajectories provide a wealth of information for phase 2 (gap acceptance). The IVD trajectory dataset was used to create models for two of these steps: Lane change desire and Lane change availability. The aspect of Lane change mechanics (i.e. how a lane change is physically made) could not be extracted from the IVD since detailed lateral positions are not contained within the IVD. Therefore, it was decided to use the IVD to build a sub model concerned with lane changing for speed advantage, from lane 2 to lane 3. In so doing, the desire to change lane for a speed advantage was assumed to be linked to the attainable speed in the current lane, and the attainable speed in the target lane.

All lane changes from lane 2 to lane 3 were extracted from the vehicle trajectory dataset, 2044 lane changes in total. This included lane changes at all sites, not just site 3 and site 4 as used in the development of the longitudinal model described earlier. Vehicles of all lengths were also considered since the numbers of lane changing vehicles was fewer than those used for the longitudinal model. Of these lane changes the speed of the vehicle immediately ahead of the subject vehicle just before a lane change, and the speed of the vehicle immediately ahead in lane 3, was captured. This effectively gave a value of the attainable speed in the current lane (as provided by the vehicle immediately ahead), and the available speed in the target lane (as provided by the speed of the vehicle immediately ahead in the target lane).

Information on lane change availability (i.e. gap acceptance) was extracted by considering the front and back headway accepted by vehicles that made a lane change from lane 2 to lane 3. A deterministic distance lead and lag gap threshold was extracted from the dataset since there was no information on the gaps that vehicles rejected, only those that were accepted (i.e. the IVD did not contain direct information on the thought process of drivers). Manual observation of the dataset suggested that a lead and lag gap between 10 and 12 metres would be appropriate. This was refined by analysing the number of lane changes greater than a certain lead or lag gap, and plotting the number of these lane changes in each case. This is shown in Figure 3 alongside the derivative for each series. There is a clear drop in gradient at about 11.5 metres; this distance was chosen as an appropriate lead and lag gap.

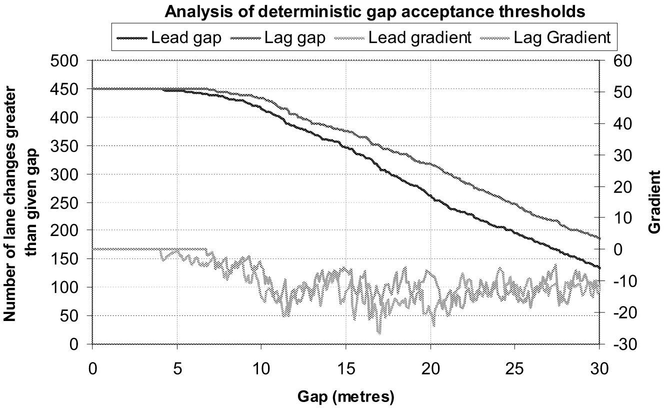


Figure 3: Analysis of Gap Acceptance

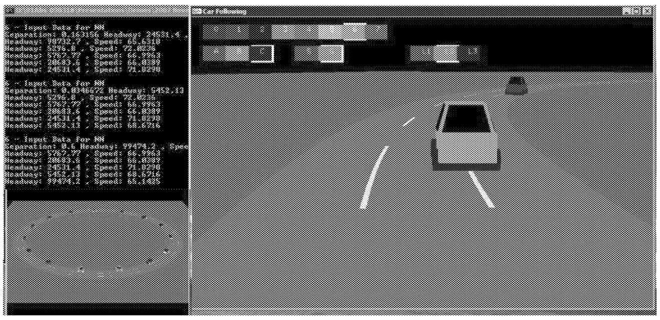
Like the longitudinal model, the speed advantage lane changing model was developed from observed lane changes, built from so many vehicle trajectories and, itself provides a unique and novel set of results.

## RESULTS AND DISCUSSION

The longitudinal and lateral models described in the previous sections were incorporated and implemented within a micro-simulation framework. A number of simulations were conducted on a 1km circular track, representing a motorway. This network allowed fundamental analysis of lateral and longitudinal movements. The following subsections describe the results from these simulations for lane changing as this provides the most interesting for a complex lateral behaviour.

### Results for multiple vehicles, with lane changing

The lateral model described previously was implemented within the micro-simulation framework. The model was built using data from vehicles moving from lane 2 to lane 3, although within the simulation was used to model movements from lane 1 to lane 2. Lane 3 was banned. The simulation also did not consider reverse movements back to lane 1. The results from the simulation runs are shown in Figure 4, with a vehicle changing from lane 1 to lane 2.



```
5 - Input Data for NN
Separation: 0.163156 Headway: 24531.4 ,
Headway: 98732.7 , Speed: 65.6310
Headway: 5296.8 , Speed: 72.8236
Headway: 5767.77 , Speed: 66.9963
Headway: 28683.6 , Speed: 66.0397
Headway: 24531.4 , Speed: 71.8290

5 - Input Data for NN
Separation: 0.0346672 Headway: 5452.13
Headway: 5296.8 , Speed: 72.8236
Headway: 5767.77 , Speed: 66.9963
Headway: 28683.6 , Speed: 66.0397
Headway: 24531.4 , Speed: 71.8290
Headway: 5452.13 , Speed: 68.6716

5 - Input Data for NN
Separation: 0.6 Headway: 99474.2 , Spee
Headway: 5767.77 , Speed: 66.9963
Headway: 28683.6 , Speed: 66.0397
Headway: 24531.4 , Speed: 71.8290
Headway: 5452.13 , Speed: 68.6716
Headway: 99474.2 , Speed: 65.1425
```

Figure 4: Lane Changing from Lane 1 (inside lane) to Lane 2 (middle lane). Circular Track Shown on Bottom Left

Results from the analysis are shown in Figure 5 below with 5 vehicles. The back vehicle maintained a constant speed of 60kph in lane 1. The results show that all four vehicles eventually overtook the slow moving vehicle in lane 1, and remained in lane 2. They all performed the lane change at different times. The vehicles changed lane quite late, and spent some time driving behind the slower vehicle. It is anticipated that in reality vehicles would change lane earlier under such circumstances, especially given the lack of vehicles in the target lane. It is recommended that, for a more refined model, lane changing desire probability distributions should be created for different levels of traffic density. Furthermore a corresponding model for movements back to the original lane should also be created.

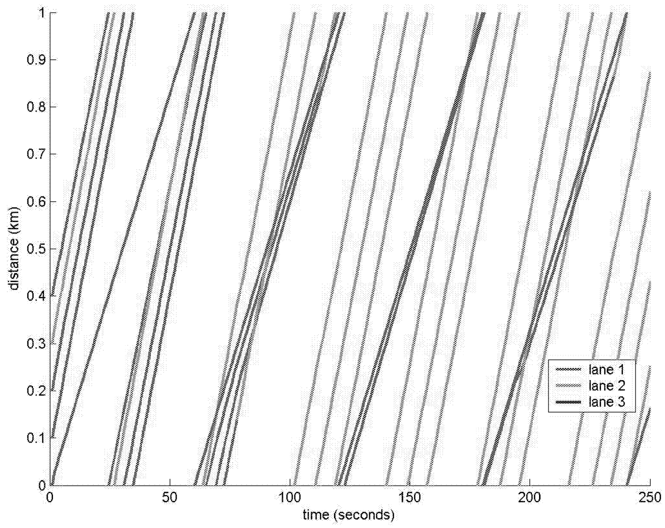


Figure 5: Lane Changing Enabled, 5 vehicles, Back Vehicle Maintained a Constant Speed of 60 kph

## CONCLUSIONS

This paper has described the work as part of a traffic micro-simulation project at TRL. The research and development has concentrated on building a model from individual vehicle trajectory data; an approach that has only until recently been adopted by model developers. Furthermore, the microscopic field data used to build the model in this report describes many more vehicles than the traditional microscopic datasets. Since TRL's IVD trajectory data does not contain vehicle behaviour in congested conditions, the models described in this paper are mainly approximate to free-flow, higher speed conditions.

The longitudinal (car following) model was considerably enhanced. A new lateral (lane changing) model was created. The development of both models has demonstrated that behavioural statistics can be extracted from the IVD trajectory dataset, and microscopic models can be built and implemented from them. The models simulate a subset of driver behaviour. The longitudinal model was calibrated using extensive measurements from car following conditions, although lacked calibration data from collision avoidance and stop-start conditions. The lateral model attempted to predict lane changes one lane to the right for a speed advantage, and did not attempt to model movements back to the original lane.

The models developed for this project are uniquely defined by its 'bottom-up' development, providing high confidence in the output of the model, particularly for scenarios that could not be validated with existing field data: it could become a leading motorway micro-simulation model.

### Current limitations and further research

The approach adopted for the longitudinal (car following) model did not attempt to vary the action point boundaries by the speed of the vehicle; an approach used in some other Action Point models. Since the original dataset did not contain a diverse range of speeds this did not seem appropriate. However, if more data were collected, across a wider range of speeds, then a study into the variation of actions points by speed could be conducted. Furthermore, the current model does not capture any variation amongst drivers and that, in reality, different drivers will behave across different action points and boundaries, even changing their boundaries over the course of a journey. This area could be investigated further both using the existing dataset, and/or with a new more comprehensive dataset (not necessarily IVD) to continue to expand the capability of the models.

The sub model is simple in that it only describes speed advantage from lane 2 to lane 3. Other types of lane changing behaviour were not extracted, like returning to an original lane, or moving out to accommodate a merging vehicle etc. With further research, additional sub models could be created and combined to make a more complete lane changing model.

## ACKNOWLEDGMENT

The authors would like to thank TRL Academy Director, Neil Paulley and his Chief Scientists for supporting this work, TRL technical reviewer of the paper, Nicholas Taylor and the anonymous referees for their useful comments on the paper. Any views expressed in it are not necessarily those of TRL or AECOM.

## REFERENCES

Ahmed, K.I; M.E. Ben-Akiva; H. N. Koutsopoulos; and R.G. Mishalani 1996. “Models of Freeway Lane Changing and Gap Acceptance”. In *Transportation and*

*Traffic Theory 1996,* J.-B Lesort (Eds.). New York: Elsevier Publishing.

Baguley, P.C.; E.J. Hardman; G.M. Lunt; and J. Quick. 2003. **"**Review of Modelling of Motorway Lane Changing". Published Project Report, PR/T/152/03, Crowthorne House

Bonsall P. 2000. *Understanding Traffic Systems - Data, Analysis and Presentation*. Ashgate, Second Edition 2000, pp. 455.

Brockfeld, E. and P. Wagner. 2006. "Validating Microscopic Traffic Flow Models", In *Intelligent Transportation Systems Conference, (ITSC),* (Toronto, Ont., Sept.17-20). INSPEC, 1604 - 1608, ISBN: 1-4244-0093-7

Chandler, R.E.; R. Herman; and E.W. Montroll. 1958. "Traffic Dynamics: Studies in Car Following". *Operations Research*, Vol. 6, pp. 165-184.

Daganzo C.F. 1999. "A Behavioral Theory of Multi-Lane Traffic Flow Part I: Long Homogeneous Freeway Sections". *Transportation Research B*, *Elsevier* Vol. 36, No 2, 131-158.

Dougherty, M; K. Fox; M. Cullip; M. Boero. 2000. "Technological Advances that Impact on Microsimulation Modelling" *Transport Reviews*, Vol 20, Issue 2, 145 - 171.

Dumbuya, A. G. Lunt; J. Weekley; A. Booth; and A. Lewis. 2007. "Neural Network Microsimulation of Motorway Traffic"-In *Industrial Simulation Conference, (ISC'07),* (Netherlands, June 2007) EUROSIS.

Dumbuya, A; G. Lunt; J. Weekley; A. Booth; and A. Lewis. 2006. "A Novel Traffic Micro-simulation Framework: Neural Network Based Modelling of Motorway Traffic Behaviour Using Individual Vehicle Data (IVD)".TRL Unpublished Project Report. Available on application to TRL.

Fritzsche H.T. 1994. "A Model for Traffic Simulation." *Traffic Engineering and Control*, May, 317–321

Gipps P.G. 1981. "A Behavioural Car-following Model for Computer Simulation." *Transportation Research, Part.-B* Vol. 15, 105-111

Gunay B. 2007. "Car Following Theory with Lateral Discomfort", *Transportation Research Part B,* Elsevier, Vol. 41, 722–735

Helly W. 1959. "Simulation of Bottlenecks in Single Lane Traffic Flow". In *Proceedings of the Symposium on Theory of Traffic Flow*, Research Laboratories, General Motors, New York: Elsevier., pp. 207-238.

Hidas P. 2004. "Modelling Vehicle Interactions in Microscopic Simulation of Merging and Weaving", (paper accepted for publication in *Transportation Research Part C: Emerging Technologies*, Elsevier Science Ltd. ISSN: 0968-090X).

Hoogendoorn, S. P; O. Saskia; and S. Marco. 2005. "Multi-Anticipative Car-Following Behavior: An Empirical Analysis", *Traffic and Granular Flow: Springer Berlin Heidelberg,* 687-697

Kometani, E. and T. Sasaki. 1959. "Dynamic Behaviour of Traffic with a Nonlinear Spacing-Speed Relationship." *In Proceedings of the Symposium on Theory of Traffic Flow*, Research Laboratories, General Motors, New York: Elsevier., pp. 105-119.

Laval, J.A. and L. Leclercq. 2007. "Microscopic Modeling of the Relaxation Phenomenon Using a Macroscopic Lane-Changing Model." *Transportation Research B*, 42, Elsevier, 511-522.

Lunt G.M. 2005. "Vehicle Re-Identification Using Induction Loop Data." In *ECTRI-FEHRLFERSI Young Researchers Seminar* 2005.

Lunt G, and A. Dumbuya. 2007. "A Novel Traffic Micro-simulation Framework: Model Enhancements Using Detailed Individual Vehicle Data (IVD) Analysis." TRL Unpublished Project Report. Available on application to TRL

Lunt G.M. and R.E. Wilson. 2003. "New Data Sets and Improved Models of Highway Traffic." *UTSG 35th Annual Conference,* Volume 2

Lin, Y; P. Tang; W.J. Zhang; and Q. Yu. 2005. "Artificial Neural Network Modelling of Driver Handling Behaviour in a Driver-Vehicle-Environment System." *International Journal of Vehicle Design,* Vol. 37, No.1. pp. 24 - 45.

Michaels R. M. 1963. "Perceptual Factors in Car Following". *Proceedings of the Second International Symposium on the Theory of Road Traffic Flow*. Vol. Paris: OECD, pp. 44-59.

Taylor, N. B; N. Bourne; S. Notley; and G. Skrobanski. 2008. "Evidence for speed-flow relationships", In *Proceedings European Transport Conference*, (Leeuwenhorst, October).

Wiedemann R. 1974. *Simulation des Straßenverkehrsflusses*, Schriftenreihe des Instituts für Verkehrswesen, University Karlsruhe, Heft 8 (in German)

## BIOGRAPHY

**DR ABS DUMBUYA** has 10 years work and research experience and is an expert in modelling and simulation. He is a Chartered Engineer and conducts technical research in the fields of synthetic traffic environments and complexity of traffic systems for government and private sector clients. He can provide expert, practical advice on a range of topics. He sits on a number of technical committees, a referee for three international journals and author of over 40 publications.

**GEORGE LUNT** has eight years experience in the research, development and application of transport models and modelling methodology. Having worked at TRL for seven years he is now working for AECOM providing innovative transport consultancy to the English Highways Agency, Local Authorities and a range of private sector clients.

# THE CHALLENGES OF ACCURATE MOBILITY PREDICTION FOR ULTRA MOBILE USERS

Jeeyoung Kim, Ahmed Helmy
Department of Computer and Information Science and Engineering
University of Florida
Gainesville, FL, USA
{jk2, helmy} @cise.ufl.edu

## ABSTRACT

How can we obtain *realistic* mobility models? Many researchers analyze mobility data based on real human behaviors in an attempt to answer this question. But in the future, will user on-line behavior change with the introduction of new mobile services and devices? In this paper, we analyze the mobility of a subset of users who are significantly more mobile than the general WLAN users; the ultra mobile users.

VoIP device users are a very good example of the ultra mobile user population, since it is a lightweight device that can be used *on-the-go*. By analyzing these traces we aim to compare the behavior of ultra mobile users to the general WLAN users. To that extent, we contrast the mobility of the WLAN users, and four carefully selected sets of ultra mobile users across various mobility metrics.

We find that prevalence for VoIP users are lower than WLAN users and also that the average and median number of access points (APs) visited are 4 to 8 times larger than that of the WLAN users indicating that the VoIP users are more mobile than general WLAN users. VoIP users are also physically more mobile, roaming a wider area.

In order to examine whether this sharp contrast in mobility affects mobile networking protocols, we compare the performance of Markov O(1), O(2), O(3) and the LZ predictors across these different set of traces. To our surprise, we find that the average prediction success rate is over 60% for general WLAN traces while the prediction success rate drops below 25% for VoIP traces. Our study strongly suggests that both mobility modeling and location prediction should be re-visited in the context of future ultra mobile users and devices.

## INTRODUCTION

Realistic modeling of user mobility is one of the most critical research areas in wireless networks. Mobility data based on real human behaviors may give us the opportunity to improve wireless and mobile services for users in many ways. Currently, several mobility models are proposed based on the analysis of real WLAN traces [1,2,5,6,9]. However, the large collection of WLAN *usage* traces seems to capture little *mobility* from the users. The average user is usually static while using the network, and exhibits a large *off time.*

In this paper, we focus on a subset of wireless users, who use wireless VoIP devices. These users leave their devices *on* most of the time and the devices are light enough to *walk and talk.* Hence, these users show a more mobile characteristic than laptop or other heavy device users while connected to the network. We aim to compare the behavior of highly mobile VoIP users to the general WLAN users by analyzing these traces. This sheds light on the realism of WLAN trace-based models. We also aim to examine the effect of any differences on protocol performance, e.g., prediction protocols.

Particularly, we compare the mobility of VoIP user traces to whole WLAN traces (as used in previous studies) and also to some test sets we have generated based on criteria that distinguish these test sets as highly mobile compared to others. We use the metrics of prevalence, number of visited APs and activity range defined in Section 3 to capture some of the main mobility characteristics of the users in our study. Our results clearly indicate that there is a significant difference between VoIP users and general mobile users, which strongly suggests revisiting mobility models for future *always-on* portable devices.

But does such dramatic contrast in mobility affect mobile networking protocols? In order to quantify such effect we examine the accuracy of several classes of mobility prediction protocols under various conditions of realistic mobility.

We compare these different sets of traces using several different predictors including the Markov O(1), O(2), O(3) and also the LZ predictor. Our experiments indicate that the Markov O(2) is the predictor with the highest accuracy among the four predictors and the LZ has the lowest. Surprisingly, all predictors perform quite poorly with VoIP users with an average of approximately 25% correct prediction rate, compared to 60% for the general WLAN users. These results prompt re-visiting of such algorithms for ultra mobile users.

We provide guidelines and pointers to improve mobility modeling and prediction protocols for highly mobile users based on the lessons learned from this study. Based on such insight we plan to develop complete solutions to these problems in our future work.

The rest of the paper is organized as follows. In Section 2, we discuss related work and approaches. In Section 3, we outline our experimental setup along with background information on our data sets and metrics. In Section 4, we examine the difference of mobility between WLAN and ultra mobile users.

In Section 5 we explore different predictors and the different prediction results between WLAN and ultra mobile users. Section 6 concludes the paper and discusses future work.

## RELATED WORK

The related work lies in the areas of mobility modeling and mobility (and location) prediction. Among the numerous modeling techniques for mobility (random, synthetic, etc.) the most realistic is the trace-based mobility modeling. Many mobility modeling techniques so far were done based on analyzing the collective WLAN traces. Model T[5] and T++[6] are empirical registration models derived from the WLAN registration patterns of the mobile users. They are able to formulate the inter-dependence of space and time explicitly by a set of few equations.

In [1], Hsu et al. proposed a mobility model to capture time-variant user mobility. In this model, they define communities that are visited often by the nodes to capture the skewed location visiting preferences, and use time periods with different mobility parameters to create the periodical re-appearance of nodes at the same location. Hsu et al. [9] also looks into modeling generic WLAN users by identifying the mobility characteristics of individual users. In [7], Balazinska et al. studied user mobility patterns and introduced metrics to model user mobility from a four week trace collected in a large corporate environment. They also analyzed user distribution and load distribution across APs. Most of these works are directly based on WLAN traces which can be found under the MobiLib project [13] or the CRAWDAD project [14].

Interestingly, only a few researches were done on the VoIP trace. Kim et al. in [4] analyzed the VoIP trace from the Dartmouth WLAN trace and tried to come up with a mobility model by analyzing pause times, speeds, paths and locations of the users. But their paper does not analyze the difference of mobility and predictability between WLAN and VoIP traces. To the best of our knowledge there has not been any work done using predictors to compare the mobility of different users.

Song et al. in [3] investigated several domain-independent predictors for the location prediction on the WLAN trace, but did not define mobility characteristics or propose any techniques to construct the mobility model. Based on the comparison result, they gave some suggestions for the usage of the predictor on WLAN traces. There are a number of user mobility prediction algorithms [10, 11] in the current literature that target cellular networks. These predictors are used in a different setting and for different purposes (i.e. paging scheme [10], efficient handoff [10], resource reservation [11]). The characteristics and scale of the predictions mentioned in the above literature are different from what we are working on. The difference including, but not limited to, the fact that a cellular device showing up in a cell that is a long distance away is very low, thus it is bounded location-wise. Whereas, in our study the mobile user could easily log off and then log back on from a totally different location at a random time. Chinchilla et al. [12] proposes usage of the Markov Chain to model the associations of users in order to improve the performance of wireless infrastructures using efficient caching, prefetching and load balancing methods on web traffic.

In our study, we use four predictors that have already been explored in existing literature [3] to verify the difference of the prediction accuracy due to mobility. The LZ predictor predicts in the case when the next symbol in the produced sequence is dependent on only its *current state*. The Order-k Markov predictor assumes that the location can be predicted from the current context which is the sequence of the k most recent symbols in the location history. The probability equation in the Markov Family considers how often the string of interest occurs in the entire input string.

## EXPERIMENTAL SETUPS

### Data Sets

We use the 3 year long Dartmouth movement trace [14] collected from 2001 to 2004 in our study. There are 13888 users and 623 different APs in this particular trace. While using this trace as our standard, general WLAN user base, we also extract other ultra mobile user data sets from this trace.

The VoIP data set we use in this work is a subset of the WLAN trace above and consists of 97 users. These are acquired by mapping the whole WLAN trace with a MAC to device type map, which is a list of all the MAC addresses mapped with the type of device it is by looking at the first three octets of the MAC addresses. In this data set we observe 2 types of VoIP devices - the Cisco7920 and Vocera devices. We have particularly chosen VoIP to measure the mobility of WLAN users since VoIP devices are always on the *on* state unlike other pocket PCs or PDAs that may easily go into hibernate mode or may even be turned on and off frequently.

Along with the VoIP data set we have generated ultra mobile test sets from the same traces in order to validate our findings. There are three test sets used in this work and they are all considered to be ultra mobile users. The 'ap_200' set is a collection of users who have visited 200 or more distinct APs and the 'ap_170' set is a collection of users who have visited more than 170 APs but less than 200 during the length of the trace. The 'range' set is a collection of users who have covered the largest physical area during the length of the trace. This was done by studying the AP location file and calculating the area range that each user has covered. Each of these test sets has approximately 100 users each. The following table 1 shows the different characteristics of the different data sets at a glance.

**Table 1: Data Sets Used from Dartmouth Trace 2001-2004**

| *Labels* | *# of users* | *Characteristics* |
|---|---|---|
| WLAN | 13439 | General WLAN users |
| VoIP | 97 | VoIP device users |
| ap_200 | 112 | Users that have visited more than 200 distinct APs |
| ap_170 | 127 | Users that have visited more than 170 less than 200 APs |
| ap_range | 113 | Users who have covered the largest physical area range |

### Metrics

How do we measure the difference of mobility that exists for different users? How do we say one user is more mobile than another? In order to answer these questions and quantify user mobility in order to compare and investigate users with different mobility levels, we come up with the following metrics: prevalence, number of visited APs and activity range.

Our evaluation metrics include prevalence, number of APs visited and the activity range. Prevalence of the traces indicates the *time that a user spends on a given AP, as a fraction of the total amount of time that they spend on the network.* The activity range comparisons is to analyze how wide an area each user has visited during each session, where the definition of session is *the time duration from the user connecting to the network to when they disconnect or disappear from the network.* The activity range is defined as *the smallest square area which can cover all the APs the user visited in an activity.* We also analyze the activity range to see what the frequency of the activity range is for different sets of users. To compare the performance of different predictors we use the prediction accuracy metrics which define the percentage of correct prediction for each user in the following section.

## MOBILITY METRICS

In our work, we compared the mobility characteristics of WLAN traces and VoIP traces from several different aspects. The evaluation metrics include prevalence, the number of APs visited by the users and the activity range where a user has been active in. The results of the comparison for each of our evaluation metrics is listed and shown as follows.

### Prevalence

Prevalence is one of the mobility metrics proposed in [7], which indicates *the time that a user spends at a given AP, as a fraction of the total amount of the time that they spend on the network.* Higher prevalence means user spent more time on such an AP, and thus less mobile. Figure 1 and 2 show that VoIP users are more mobile than WLAN users, since the bar is lower. Especially for the most right bar which indicates prevalence higher than 0.95, the WLAN is much higher than VoIP. This means there are larger portion of users in WLAN who spent most of their time on only one AP than that in VoIP.

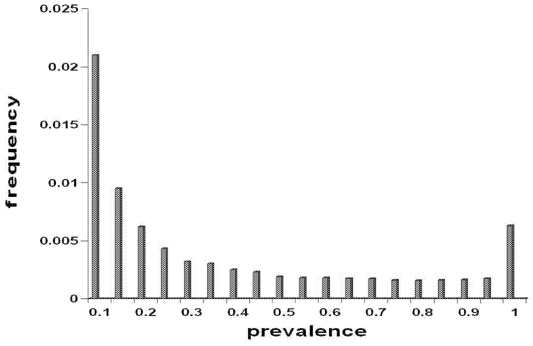


Figure 1: Prevalence of WLAN user trace

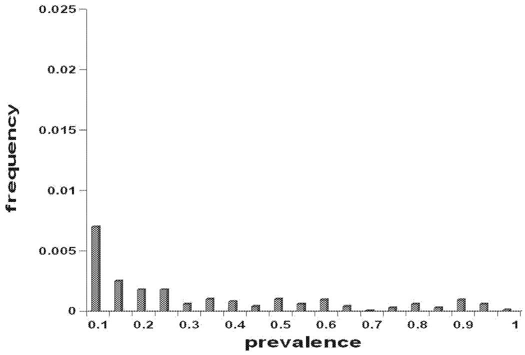


Figure 2: Prevalence of VoIP device user set

### Number of APs Visited

Figure 3 and 4 shows the number of APs visited distribution CDF by WLAN and VoIP users. This clearly shows VoIP users visited much more APs than WLAN users. The average number of APs that the VoIP users visited is about 4.1 times than that of the WLAN users while the median number of APs that the VoIP users visited is about 7.7 times than that of WLAN.

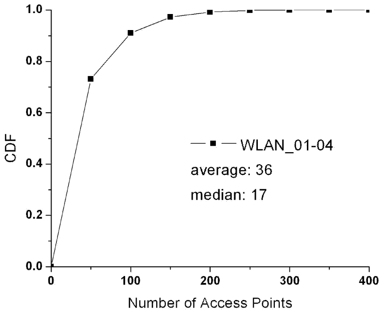


Figure 3: Number of APs visited in the whole WLAN trace

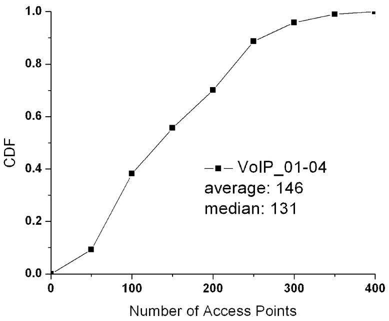


Figure 4: Number of APs visited in the VoIP device user set

### Activity Range

Activity range is defined as the smallest square area which can cover all the APs the user has visited in a given activity. Figure 5 and 6 shows the activity range distribution for WLAN and VoIP users. The percentage of VoIP users having a larger area of activity range is higher than that of the WLAN users who, most of the time tends to stay in a very limited area.

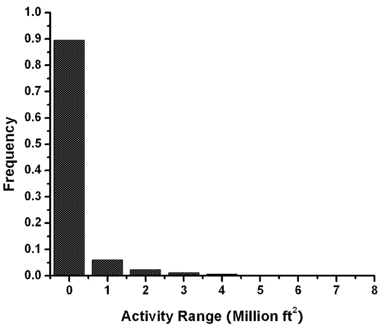


Figure 5: Activity Range Distribution of WLAN trace

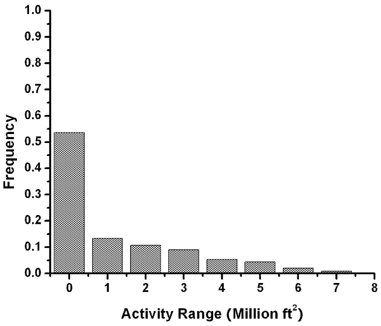


Figure 6: Activity Range Distribution of VoIP device users

## PREDICTABILITY COMPARISON

To study the effect of the sharp contrast in mobility and behavioral characteristics between VoIP and other WLAN users on networking protocols we analyze a set of well known prediction algorithms with the various sets of traces we have in our study.

We have run the Markov O(1), O(2) and O(3) predictors along with the LZ [3] predictor for each of the test sets we have, and also for the VoIP trace set and the whole body of the WLAN trace. We also compared the accuracy of all four predictors with the VoIP trace data to see which one has the best performance. Accuracy is measured as percentage of correct predictions of the next AP to visit. As shown in figures 7 through 12, the ***WLAN trace always had the best prediction accuracy for all the predictors with an average of about 60% accuracy. The VoIP trace, by contrast, had the worst prediction accuracy for all of the predictors with an average of approximately 25% accuracy.*** **From these graphs we see that the best accuracy can be no more than 80% for VoIP users, while more than 95% accuracy for WLAN users.**

When we were first conducting our experiment, we expected that the range of the physical area that each user covered would be a better criterion to measure mobility than the number of APs visited since we consider a person to be more mobile when that person covers more ground. Hence, we expected that the 'range' set would return very bad prediction accuracy. Surprisingly, the 'range' set always exhibits performance between of the other two test sets (ap_200 and ap_170), which indicates that the users that covered larger areas physically most likely have visited an average of 200 APs during their lifetime.

To explain this result, intuitively the users that had visited less APs also had a better prediction rate than that of the users who had visited more APs. The difference of the prediction accuracy between the two test sets is always around 10% near the median.

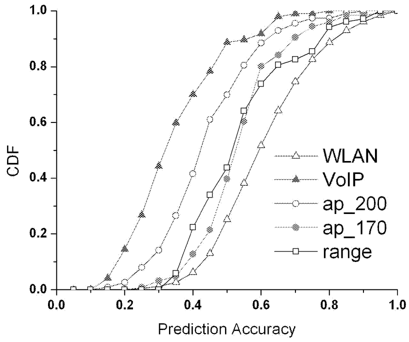


Figure 7: Prediction Accuracy of Markov O(1) Predictor

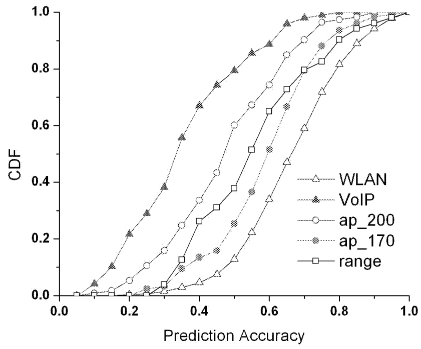


Figure 8: Prediction Accuracy of Markov O(2) Predictor

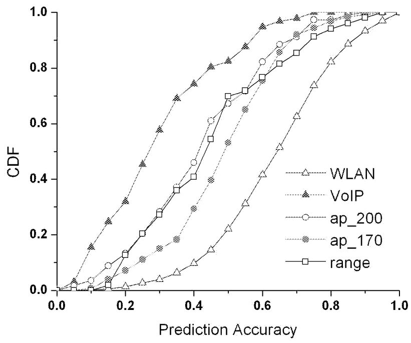


Figure 9: Prediction Accuracy of Markov O(3) Predictor

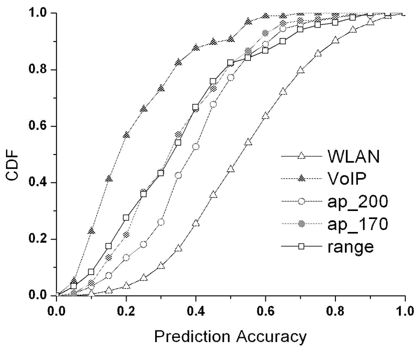


Figure 10: Prediction Accuracy of LZ Predictor

As for the comparison of the predictors on the VoIP data set and WLAN trace, the LZ predictor showed the worst prediction rate and the Markov O(2) showed the best prediction accuracy by a very minimal difference from the Markov O(1). Markov O(3) did not show a good prediction and these results indicate that a larger data structure and higher complexity does not help in making better predictions. However, the four predictors that are used in this work do *not* provide good prediction for the VoIP data set, although they are showing a very similar trend regardless of the mobility of the user.

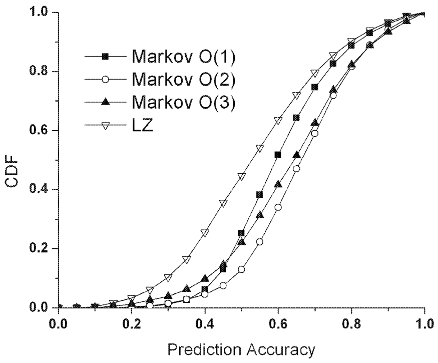


Figure 11: Comparison of Predictability on VoIP device users

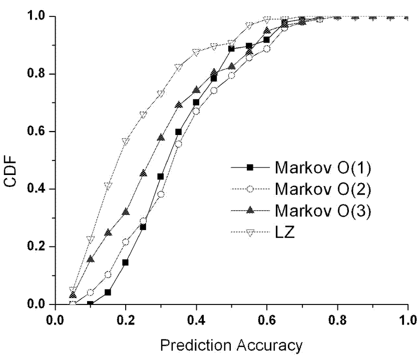


Figure 12: Comparison of Predictability on VoIP device users

## CONCLUSION AND FUTURE WORK

Our findings open the door for revisiting mobility modeling and improved prediction of ultra mobile users. We can see from our findings that whatever protocols and services (i.e. prediction) that were developed for the normal WLAN user can change dramatically in the environment of future ultra mobile users. We plan to design a better predictor for "ultra mobile" users. Our plan includes investigating *domain-specific* knowledge, regressions, schedules and repetitive or preferential user behavior. The success rate should also be taken into consideration since depending on the granularity of the success rate the prediction accuracy may be highly affected. We shall also examine the adequacy of WLAN trace based mobility models for ultra mobile and VoIP users, that are likely to increase in the future.

## REFERENCES

[1] W. Hsu, T. Spyropoulos, K. Psounis and A. Helmy, *"Modeling Time-variant User Mobility in Wireless Mobile Networks"*, Proceedings of IEEE Conference on Computer Communications, May 2007, Anchorage, Alaska

[2] C. Tuduce, T. Gross, *"A mobility model based on WLAN traces and its validation"*, Proceedings of INFOCOM 2005: Miami, FL, USA 664-674

[3] L. Song, D. Kotz, R. Jain and X. He, "Evaluating location predictors with extensive Wi-Fi mobility data", Proceedings of IEEE INFOCOM 2004, Hong Kong, China

[4] M. Kim, D. Kotz, and S. Kim, *"Extracting a mobility model from real user traces"*, Proceedings of IEEE INFOCOM 2006, Barcelona, Spain

[5] R. Jain , D. Lelescu , M. Balakrishnan, *"Model T: an empirical model for user registration patterns in a campus wireless LAN"*, Proceedings of the 11th annual international conference on Mobile computing and networking, August 28September 02, 2005, Cologne, Germany

[6] D. Lelescu , U. Kozat , R. Jain , M. Balakrishnan, *"Model T++:: an empirical joint space-time registration model"*, Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing, May 22-25, 2006, Florence, Italy

[7] M. Balazinska and P. Castro, "*Characterizing Mobility and Network Usage in a Corporate Wireless Local-Area Network*" in International Conference on Mobile Systems, Applications, and Services, May 2003

[8] T. Henderson, D. Kotz, I. Abyzov, *"The changing usage of a maturecampus-wide wireless network"*, Proceedings of the MOBICOM 2004: 187-201

[9] W. Hsu and A. Helmy, *"On Modeling User Associations in Wireless LAN Traces on University Campuses"* The Second International Workshop on Wireless Network Measurement (WiNMee 2006), Boston MA, Apr. 2006

[10] H. Zang and J. C. Bolot, *"Mining Call and Mobility Data to Improve Paging Efficiency in Cellular Networks"*, Proceedings of the MOBICOM 2007: 123-134

[11] J. Chan and A. Seneviratne, *"A Practical User Mobility Prediction Algorithm for Supporting Adaptive QoS in Wireless Networks",* Proceedings of IEEE International Conference on Networks, Sep. 1999, Brisbane

[12] F. Chinchilla, M.Lindsey, M Papadopouli, "Analysis of Wireless Information locality and association patterns in a campus", Proceedings of the IEEE INFOCOM 2004, Hongkong, China

[13] http://nile.cise.ufl.edu/MobiLib/

[14] http://crawdad.cs.dartmouth.edu/

# MANAGING SPATIAL SELF-ORGANIZATION VIA COLLECTIVE BEHAVIORS

Rawan Ghnemat(1), Cyrille Bertelle(1), Gérard H.E. Duchamp(2)

(1)LITIS - University of Le Havre, 25 rue Philippe Lebon, BP 540, 76058 Le Havre Cedex, France
email: rawan.ghnemat@litislab.eu, cyrille.bertelle@litislab.eu

(2)LIPN - University of Paris XIII, 99 avenue Jean-Baptiste Clément, 93430 Villetaneuse, France
email: ghed@lipn.univ-paris13.fr

## KEYWORDS

swarm intelligence, complex systems, self-organization, ant systems, Schelling model, spatial organization

## ABSTRACT

Spatial self-organizations appear in many natural and artificial systems. Spatial systems creation and development, called morphogenesis, is the subject of many research studies since many years (1). Fractal computation approach is, for exemple, one of the methods proposed to deal with such studies. But, even if this method is able to describe unlimited local formations on multi-scale descriptions, the formation process itself is described in a global way. The goal of this paper is to introduce the distributed and decentralized computing as a general methodology to propose emergent spatial formation, able to deal with local perturbations and with non homegeneous formation rules. We propose a study case on Schelling Model dealing with interacting population over an environment based on a regular grid.

## SPATIAL MORPHOLOGY MODELLING ON THE EDGE OF COMPLEXITY

The study of spatial morphology is a major aspect of the understanding of many phenomena for natural or artificial systems. Living systems or social systems, for example, are systems where the spatial formation has a high meaning and modifies deeply by itself the system evolution. The system evolution leads to modify itself the spatial formations by feed-back processes.

Spatial morphology models can be classified by many criteria. Some of these models are static (finding the optimal shape of some problem) or dynamic (morphogenesis, for example). When the models involve dynamical processes, these dynamics can be expressed in a global way, like we do using partial differential equations: the objective of the system description consists in describing the different phenomena involved (diffusion, transport, ...).

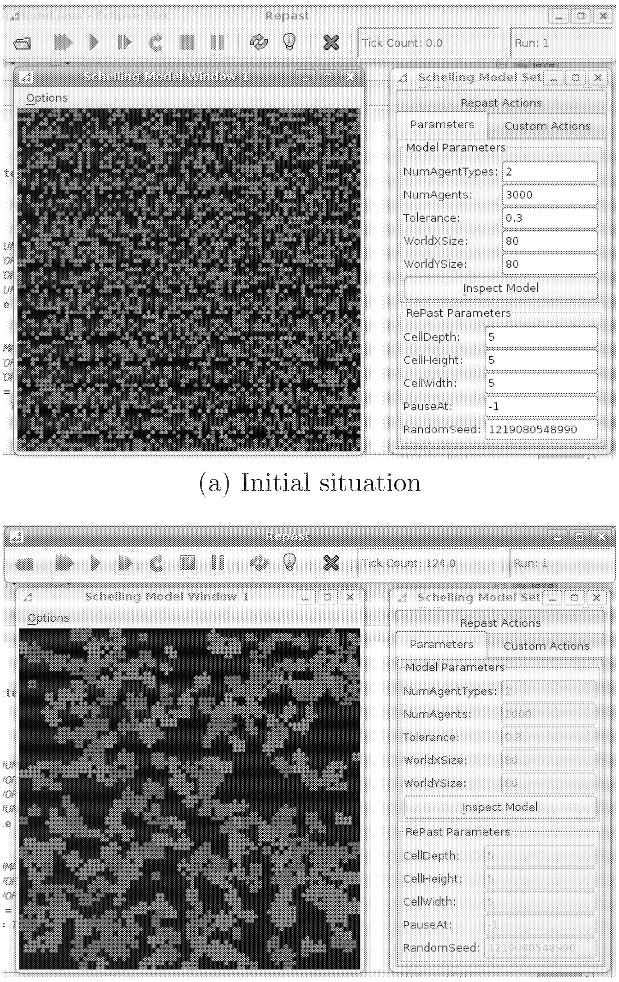
Spatial morphology systems can be involved inside a multi-scale processus, giving some specific properties to this multi-scale formation, like the development of important exchange area. Fractal systems are well-known to model these multi-scale systems like fractal shape of plants.

Even if these fractal geometries are able to model multi-scale descriptions, they are generally completly deterministic and they are not suitable to describe geometrical evolutions or to integrate local disturbations. For this purpose, we need to change the model concept and go from global deterministic models to decentralized approaches where the whole system is only known (or emerge) by the interaction system of behavior population.

## SCHELLING MODEL EXTENSION IN ORDER TO MODEL SELF-ORGANIZATION BY MEANING OF MULTI-CRITERIA SYSTEM SEPARATION

Thomas Schelling's city segregation model illustrates how spatial organizations can emerge from local rules, concerning the spatial distribution of people which belong to different classes. In this model, people can move, depending on their own satisfaction to have neighbours of their own class. Based on this model, a city can be highly segregated even if people have only a mild preference for living among people similar to them.

In this model, each person is an agent placed on a 2D grid (in his original presentation, a chessboard was used by Thomas Schelling). Each case can be considered like a house where the agent lives. Each agent cares about the class of his immediate neighbours who are the occupants of the abutting squares of the chessboard. Each agent has a maximum of eight possible neighbours. He computes the rate of the neighbours of its own class from its eight possible neighbours. Each agent has a tolerence rate determining whether he is happy or not at his current house location. If the rate of the neighbours of its own class is under this tolerence rate, he decides to move to live in another free place in the 2D grid.



Tick Count: 0.0
Run: 1
Model Parameters
NumAgentTypes: 2
NumAgents: 3000
Tolerance: 0.3
WorldXSize: 80
WorldYSize: 80
RePast Parameters
CellDepth: 5
CellHeight: 5
CellWidth: 5
PauseAt: -1
RandomSeed: 1219080548990

(a) Initial situation

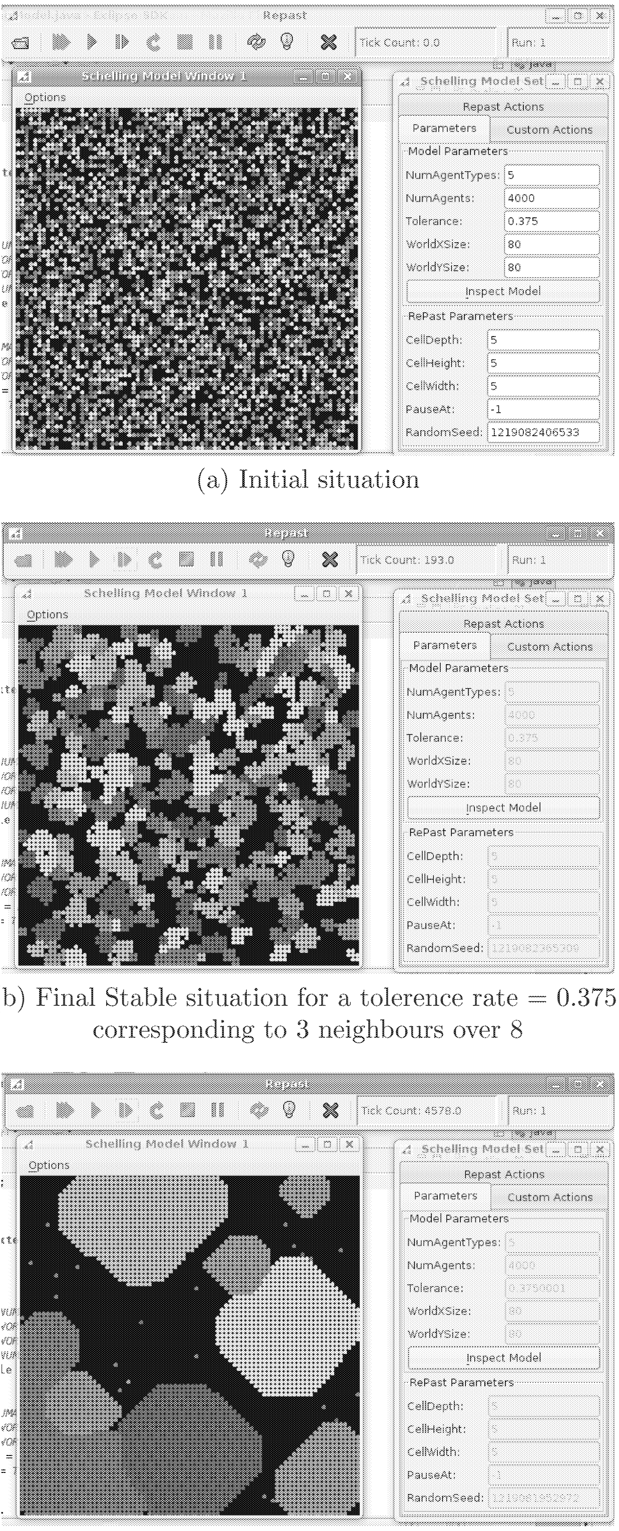
Tick Count: 124.0
Run: 1
Model Parameters
NumAgentTypes: 2
NumAgents: 3000
Tolerance: 0.3
WorldXSize: 80
WorldYSize: 80
RePast Parameters
CellDepth: 5
CellHeight: 5
CellWidth: 5
PauseAt: -1
RandomSeed: 1219080548990

(b) Stable situation after 124 iterations

Figure 1: Standard 2 populations seggregation Schelling model on RePast

The exact degree of segregation which emerges in the city depends strongly on the specification of the agents tolerance rate. It is noticeable that, under some rule specifications, Schelling's city can transit from a highly integrated state to a highly segregated state in response to a small local disturbance. We can observe some bifurcation phenomena which lead to chain reaction of displacements.

In figure 1, we show an implementation of this algorithm using the multi-agent platform called Repast (7). Here, we show the "classical" and original problem, modelling the segregation phenomenon with two population classes, described by red and blue squares. Both,



Tick Count: 0.0
Run: 1
Model Parameters
NumAgentTypes: 5
NumAgents: 4000
Tolerance: 0.375
WorldXSize: 80
WorldYSize: 80
RePast Parameters
CellDepth: 5
CellHeight: 5
CellWidth: 5
PauseAt: -1
RandomSeed: 1219082406533

(a) Initial situation

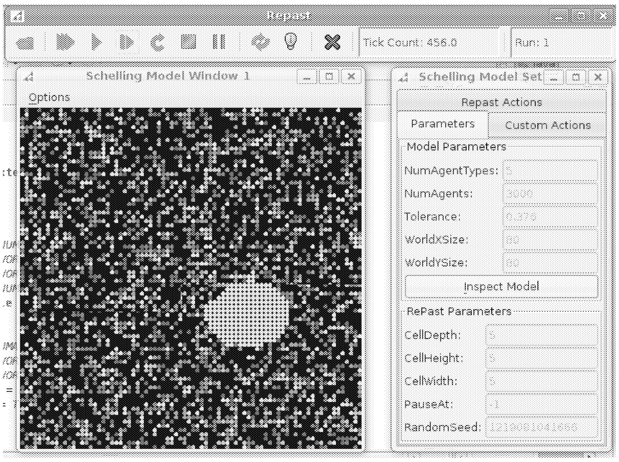
Tick Count: 193.0
Run: 1
Model Parameters
NumAgentTypes: 5
NumAgents: 4000
Tolerance: 0.375
WorldXSize: 80
WorldYSize: 80
RePast Parameters
CellDepth: 5
CellHeight: 5
CellWidth: 5
PauseAt: -1
RandomSeed: 1219082365309

(b) Final Stable situation for a tolerence rate = 0.375, corresponding to 3 neighbours over 8

Tick Count: 4578.0
Run: 1
Model Parameters
NumAgentTypes: 5
NumAgents: 4000
Tolerance: 0.3750001
WorldXSize: 80
WorldYSize: 80
RePast Parameters
CellDepth: 5
CellHeight: 5
CellWidth: 5
PauseAt: -1
RandomSeed: 1219081952872

(c) Final Stable situation for a tolerence rate greater than 0.375 and corresponding to 4 neighbours.

Figure 2: 5 populations seggregation Schelling model on RePast with density = 0.625 and with 2 values of tolerance rate



Tick Count: 456.0

Run: 1

Model Parameters

NumAgentTypes: 5

NumAgents: 3000

Tolerance: 0.376

WorldxSize: 80

WorldYSize: 80

RePast Parameters

CellDepth: 5

CellHeight: 5

CellWidth: 5

PauseAt: -1

RandomSeed: 1219081041666

Figure 3: Singular situation for 5 populations seggregation Schelling model with density = 0.47 and with tolerance rate = 0.376

the initial population distribution and the final stable distribution are given.

In figure 2, we detail the impact of tolerence rate on the segregation result. In this figure, we extend the original problem based on 2 population classes to 5 classes population. Part (a) describes the initial distribution according to a whole population density equal to 0.625. Part (b) describes the stable population distribution for a similar tolerance rate for each agent, equal to 0.375, corresponding to 3 neighbours on 8. This value is a *discrete* bifurcation point from where all small additional value leads to a very different distribution. To illustrate this phenomenon, part (c) describes the population distribution for a tolerance rate greater than 0.375, corresponding to 4 neighbours. The final population distribution is completly different than the one in part (b).
Figure 3 describe some singular formation which can appears in very few cases, when we go over the bifurcation point which leads to no global clustering formation, except if some very small cluster kernels appear according to stochatic move spatial conjonction.

## CONCLUSION

In this paper, we discuss about the properties of decentralized methods to model the spatial self-organization. These methods are based on the description of the whole system by the interaction systems of individual behaviors. We present some experiments on Schelling's model where auto-organization emerge from local rules. This model is able to take into account multi-criteria : each population class could be understand as the characteristic of some specific criteria. And the processus leads to simulate the emergent system morphology in order to define its criteria-specific sub-population separation (as an extension of segregation problem toward multi-component morphology development).

## REFERENCES

[1] Bourgine, P. and A. Lesne (eds) (2006) *Morphogenèse : L'origine des formes*, "Echelles" series, Belin.

[2] Benenson, I. and P.M. Torrens (2004) *Geosimulation - Automata-based modeling of urban phenomena*, Wiley.

[3] Bertelle,C.; Duchamp, G.H.E. and H. Kadri-Dahmani (eds) (2008) *Complex Systems and Self-Organization Modelling*, "Understanding Complex Systems" series, Springer, (in press).

[4] Bonabeau, E.; Dorigo, M. and G. Theraulaz (1999) *Swam Intelligence, from natural to artificial systems*, "Santa Fe Institute Studies in the Sciences of Complexity" series, Oxford University Press.

[5] Ghnemat, R.; Bertelle, C. and G.H.E Duchamp (2007) *Adaptive Automata Community Detection and Clustering, a generic methodology*, in *Proceedings of World Congress on Engineering 2007, International Conference of Computational Intelligence and Intelligent Systems*, pp 25-30, London, U.K., 2-4th July 2007.

[6] Kennedy, J. and R.C. Eberhart (1995) *Particle Swarm Optimization* In *Proceedings of IEEE International Conference on Neural Networks (Perth, Australia)*, IEEE Service Center, Piscataway, NJ, 5(3), pp 1942-1948.

[7] Repast web site (2008)
`http://repast.sourceforge.net`.

[8] Reynolds, C.W. (1987) *Flocks, Herds and Schools: a distributed behavioral model* In *Computer Graphics*, 21(4) (SIGGRAPH'87 Conference Proceedings), pp 25-34.

[9] Schweitzer, F. (2003) *Brownian Agents and Active Particles*, Springer.

[10] Weiss, G. (ed.) (1999) *Multiagent Systems*, MIT Press.

[11] Xiao, N. (2005) *Geographic optimization using evolutionary algorithms* In *8th International Conference on GeoComputation*, University of Michigan, USA.

# A CONSTRICTION FACTOR BASED PARTICLE SWARM OPTIMIZATION FOR ECONOMIC DISPATCH

Shi Yao Lim, Mohammad Montakhab, and Hassan Nouri
Bristol Institute of Technology,
University of the West of England,
Frenchay Campus,
Coldharbour Lane, Bristol, BS16 1QY, United Kingdom.
E-mail: limshiyao@gmail.com

**KEYWORDS**
Constriction factor, economic dispatch (ED), non-smooth optimization, particle swarm optimization (PSO).

## ABSTRACT

This paper presents an efficient method for solving the non-smooth economic dispatch (ED) problem with valve-point effects, by introducing a constriction factor into the original particle swarm optimization (PSO) algorithm. The proposed constriction factor based particle swarm optimization (CFBPSO) combines the original PSO algorithm with a constriction factor. The application of a constriction factor into PSO is a useful strategy to ensure convergence of the particle swarm algorithm. To verify the feasibility and performance, the proposed method is applied to a test non-smooth ED problem with valve-point effects. The results of the CFBPSO are compared with the results of other methods; the genetic algorithm (GA); the evolutionary programming (EP); the modified PSO; and in particular the improved PSO (IPSO).

## 1. INTRODUCTION

Under the new deregulated electricity industry, power utilities try to achieve high operating efficiency to produce cheap electricity. High operating efficiency minimizes the cost of a kilowatt-hour to a consumer and the cost to the company delivering a kilowatt-hour in the face of constantly rising prices for fuel, labor, supplies, and maintenance. Operational economics involving power generation and delivery can be subdivided into two parts. Economic dispatch (ED), as one part is called, has the distribution of generated power at lowest cost as its main objective. Minimum-loss, as the second part is called, deals with minimum-loss delivery of the generated power to the loads.

The ED of power generating units has always occupied an important position in the electric power industry. ED is a computational process where the total required generation is distributed among the generation units in operation, by minimizing the selected cost criterion, subject to load and operational constraints. For any specified load condition, ED determines the power output of each plant (and each generating unit within the plant) which will minimize the overall cost of fuel needed to serve the system load (Wood and Wollenberg 1996). ED is used in real-time energy management power system control by most programs to allocate the total generation among the available units, unit commitment, etc. ED focuses upon coordinating the production cost at all power plants operating on the system.

In the traditional ED problem, the cost function for each generator has been approximately represented by a single quadratic function and is solved using mathematical programming based optimization techniques such as lambda-iteration method, gradient-based method, etc. (Park et al. 2006). These methods require incremental fuel cost curves which are piece-wise linear and monotonically increasing to find the global optimal solution. Unfortunately, the input-output characteristics of generating units are inherently highly non-linear due to valve-point loadings. Thus, the practical ED problem with valve-point effects is represented as a non-smooth optimization problem with equality and inequality constraints. This makes the problem of finding the global optimum solution challenging. Dynamic programming (DP) method (Liang  and Glover 1992) is one of the approaches to solve the non-linear and discontinuous ED problem, but it suffers from the problem of "curse of dimensionality" or local optimality. In order to overcome this problem, several alternative methods have been developed such as evolutionary programming (EP) (Yang et al. 1996), genetic algorithm (GA) (Walters and Sheble 1993), tabu search (Lin et al. 2002), neural network  (Lee 1998), and particle swarm optimization (Eberhart and Shi, 2000; Shi and Eberhart, 2001).

Particle swarm optimization (PSO), being one of the evolutionary computation techniques, is one of the most powerful methods for solving global optimization problems. The PSO method is a member of the wide category of swarm intelligence methods, which was first introduced by American social psychologist James Kennedy and electrical engineer Russel C. Eberhart in 1995 (Kennedy and Eberhart 1995). It was first inspired by social behavior of organisms such as fish schooling and bird flocking resulted in the possibilities of utilizing this behavior as an optimization tool (Gaing 2003). In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particle. Every particle finds its personal best position and the group best position through iteration, and then modifies their progressing direction and speed to reach the optimized position quickly. Because of the rapid convergence speed of PSO, it has been successfully applied in many areas.

Since its introduction, PSO has attracted much attention from researchers around the world. Many researchers have indicated that the PSO often converges significantly faster to the global optimum but has difficulties in premature convergence, performance and the diversity loss in optimization process. Clerc, in his study on stability and convergence of PSO has indicated that use of a constriction factor may be necessary to insure convergence of the particle swarm algorithm (Clerc 1999). His research indicated that the inclusion of properly defined constriction coefficients increases the rate of convergence; further, these coefficients can prevent explosion and induce particles to converge on local optima.

In this paper, a novel approach is proposed to the non-smooth ED problem with valve-point effects using a constriction factor based PSO (CFBPSO). The proposed CFBPSO combines the original PSO algorithm with a constriction factor. The application of a constriction factor into PSO is a useful strategy to ensure convergence of the particle swarm algorithm. Unlike other evolutionary computation methods, the proposed method ensures the convergence of the search procedure based on the mathematical theory. In order to verify the feasibility, the proposed method is tested on a three-generator power system and the results are compared with those of other methods and in particular the improved PSO (IPSO) (Park et al. 2006) in order to demonstrate its performance. The results indicate the applicability of the proposed CFBPSO method to the practical ED problem.

The rest of the paper is structured as follows. The ED problem formulation is described in Section 2. In Section 3, the proposed CFBPSO algorithm for solving the ED problem is explained. Section 4 presents the simulation results and comparison with those of other methods. Finally, in Section 5, conclusions are drawn, based on the results found from the simulation analyses in Section 4.

## 2. ECONOMIC DISPATCH FORMULATION

### *A. Basic Economic Dispatch Formulation*

Fig. 1 shows the configuration that will be studied in this section. This system consists of $N$ generating units connected to a single busbar serving a received electrical load, $P_{load}$. The input to each unit, shown as $F_i$ represents the cost rate of the unit $i$. The output of each unit, $P_i$ is the electrical power generated by that particular unit. The total cost rate of this system is the sum of costs of each of the individual units. The essential constraint on the operation of this system is that the sum of the output powers must equal the load demand. It is obvious, in the case that the generators are connected to the same busbar, the losses are zero, but if the generators are located in distant geographic locations, the losses will not be zero or close to zero. For simplicity, in this phase of the research, we assume that the losses are zero.

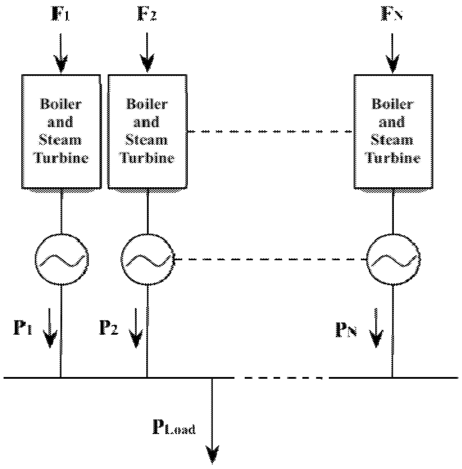


Fig. 1: *N* generating units committed to serve a load.

Mathematically speaking, the problem may be stated very concisely. That is, an objective function $F_T$, is equal to the total cost for supplying the indicated load. The problem is to minimize $F_T$ subject to the operating constraints of the power system. .

The objective of the ED problem is to minimize the total cost of generation under various system and operational constraints while satisfying the power demand. The primary concern of an ED problem is the minimization of its objective function. The total cost generated that meets the demand and obeys all other constraints associated is selected as the objective function. In general, the ED problem can be formulated mathematically as a constrained optimization problem with an objective function of the form:

$$F_T = \sum_{i=1}^{N} F_i(P_i) \tag{1}$$

where $F_T$ is the total generation cost; $N$ is the total number of generating units; $F_i$ is the power generation cost function of the $i$th unit.

Generally, the fuel cost of a thermal generation unit is considered as a second order polynomial function

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 \tag{2}$$

where $P_i$ is the power of the $i$th generating unit;
$a_i$, $b_i$, $c_i$ are the cost coefficients of the $i$th generating unit.

This model is subjected to the following constraints.

*1) Real Power Balance Equation*

For power balance, an equality constraint should be satisfied. The total generated power should be the same as total load demand plus the total line loss

$$\sum_{i=1}^{N} P_i = P_{Demand} + P_{Loss} \tag{3}$$

where $P_{Demand}$ is the total system demand and $P_{Loss}$ is the total line loss. For simplicity, transmission loss is not considered in this paper (i.e., $P_{Loss} = 0$).

*2) Unit Operating Limits*

There is a limit on the amount of power which a generator can deliver. The power output of any generator should not exceed its rating nor should it be below that necessary for stable operation. Generation output of each generator should lie between maximum and minimum limits. The corresponding inequality constraints for each generator are

$$P_{i,\min} \le P_i \le P_{i,\max} \tag{4}$$

where $P_i$ is the output power of generator $i$; $P_{i,\min}$ and $P_{i,\max}$ are the minimum and maximum power outputs of generator $i$, respectively.

### B. Economic Dispatch with Valve-Point Effect

The generating units with multi-valve steam turbines exhibit a greater variation in the fuel-cost functions (Park et al. 2006). The valve opening process of multi-valve steam turbines produces a ripple-like effect in the heat rate curve of the generators. These "valve-points" are illustrated in Fig. 2.

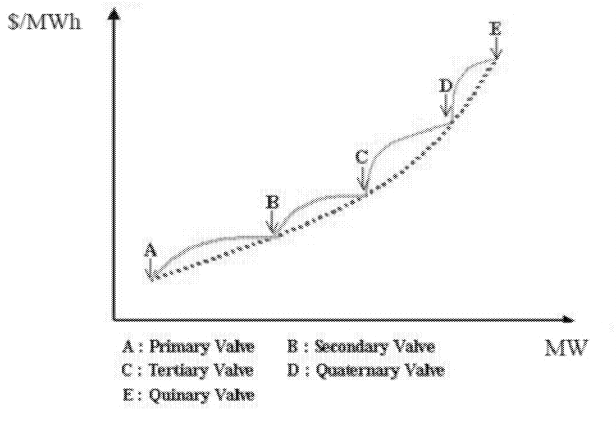


Fig. 2: Incremental fuel cost versus power output for a 5 valve steam turbine unit.

Fig. 2 shows that the cost curve increases at a greater rate with power production just as a valve is opened. The reason for this is that the so-called throttling losses due to gaseous friction around the valve-edges are greatest just as the valve is opened and taper off as the valve opening increases and the steam flow smoothens.

The significance of this effect is that the actual cost curve function of a large steam plant is not continuous but more important it is non-linear. Thus, the cost function contains higher order nonlinearity. To model the effects of "valve-points", a recurring sinusoid contribution is added to the cost function. Therefore, equation (2), the basic quadratic equation for the fuel cost of a thermal generation unit should be replaced by equation (5) to consider the valve-point effects. The valve-point effects are taken into consideration in the ED problem by superimposing the basic quadratic fuel-cost characteristics with the rectified sinusoid component as follows:

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 + \left| e_i \times \sin\left(f_i \times \left(P_{i,\min} - P_i\right)\right)\right| \tag{5}$$

where $e_i$ and $f_i$ are the coefficients of generator $i$ reflecting valve-point effects. Note that ignoring valve-point effects, some inaccuracy would be introduced into the resulting dispatch.

## 3. CONSTRICTION FACTOR BASED PARTICLE SWARM OPTIMIZATION

### A. Basic Concept of Particle Swarm Optimization

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Kennedy and Eberhart in 1995, discovered through simplified social model simulation (Kennedy and Eberhart 1995). It stimulates the behaviors of bird flocking involving the scenario of a group of birds randomly looking for food in an area. All the birds don't know where the food is located, but they just know how far they are from the food location. So, an effective strategy for the bird to find food is to follow the bird which is nearest to the food. PSO is motivated from this scenario and is developed to solve complex optimization problems.

In the conventional PSO, suppose that the target problem has $n$ dimensions and a population of particles, which encode solutions to the problem, move in the search space in an attempt to uncover better solutions. Each particle has a position vector of $X_i$ and a velocity vector $V_i$. The position vector $X_i$ and the velocity vector $V_i$ of the $i$th particle in the $n$-dimensional search space can be represented as $X_i = (x_{i1}, x_{i2}, ..., x_{in})$ and $V_i = (v_{i1}, v_{i2}, ..., v_{in})$, respectively. Each particle has a memory of the best position in the search space that it has found so far $(Pbest_i)$, and knows the best location found to date by all the particles in the swarm $(Gbest)$. Let $Pbest_i = \left(x_{i1}^{Pbest}, x_{i2}^{Pbest}, ..., x_{in}^{Pbest}\right)$ and $Gbest = \left(x_1^{Gbest}, x_2^{Gbest}, ..., x_n^{Gbest}\right)$ be the best position of the individual $i$ and all the individuals so far, respectively. At each step, the velocity of the $i$th particle will be updated according to the following equation in the PSO algorithm:

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 \times \left(Pbest_i^k - X_i^k\right) + c_2 r_2 \times \left(Gbest^k - X_i^k\right) \tag{6}$$

where,

| | |
|---|---|
| $V_i^{k+1}$ | velocity of individual $i$ at iteration $k+1$, |
| $V_i^k$ | velocity of individual $i$ at iteration $k$, |
| $\omega$ | inertia weight parameter, |
| $c_1$, $c_2$ | acceleration coefficients, |
| $r_1$, $r_2$ | random numbers between 0 and 1, |
| $X_i^k$ | position of individual $i$ at iteration $k$, |
| $Pbest_i^k$ | best position of individual $i$ at iteration $k$, |
| $Gbest^k$ | best position of the group until iteration $k$. |

In this velocity updating process, the acceleration coefficients $c_1$, $c_2$ and the inertia weight $\omega$ are predefined and $r_1$, $r_2$ are uniformly generated random numbers in the range of [0, 1]. In general, the inertia weight $\omega$ is set according to the following equation:

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{Iter_{\max}} \times Iter \qquad (7)$$

where,

| | |
|---|---|
| $\omega_{\max}$, $\omega_{\min}$ | initial and final inertia parameter weights, |
| $Iter_{\max}$ | maximum iteration number, |
| $Iter$ | current iteration number. |

The model using (7) is called the "inertia weight approach (IWA)" (Kennedy and Eberhart 2001). Using the above equations, diversification characteristic is gradually decreased and a certain velocity, which gradually moves the current searching point close to $Pbest$ and $Gbest$ can be calculated. Each individual moves from the current position (searching point in the solution space) to the next one by the modified velocity in (6) using the following equation:

$$X_i^{k+1} = X_i^k + V_i^{k+1} \qquad (8)$$

*B. Constriction Factor Approach*

After Kennedy and Eberhart proposed the original particle swarm, a lot of improved particle swarms were introduced. Because PSO originated from efforts to model social systems, a thorough mathematical foundation for the methodology was not developed at the same time as the algorithm. Within the last few years, a few attempts have been made to begin to build this foundation. The particle swarm with constriction factor is very typical. Clerc (Clerc 1999) in his study on stability and convergence of PSO have introduced a constriction factor $K$. Clerc indicates that the use of a constriction factor may be necessary to insure convergence of the particle swarm algorithm. He had established some mathematical foundation to explain the behavior of a simplified PSO model in its search for an optimal solution.

The basic system equations of the PSO (6-8) can be considered as a kind of difference equations. Therefore, the system dynamics, namely, the search procedure, can be analyzed by the Eigen value analysis and can be controlled so that the system has the following features.

a) The system converges,
b) The system can search different regions efficiently by avoiding premature convergence.

In order to insure convergence of the PSO algorithm, the velocity of the constriction factor based approach can be expressed as follows:

$$V_i^{k+1} = K\left[V_i^k + c_1 r_1 \times \left(Pbest_i^k - X_i^k\right) + c_2 r_2 \times \left(Gbest^k - X_i^k\right)\right] \qquad (9)$$

$$K = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|},\ where\ \varphi = c_1 + c_2,\ \varphi > 4 \qquad (10)$$

The convergence characteristic of the system can be controlled by $\varphi$. In the constriction factor approach, the $\varphi$ must be greater than 4.0 to guarantee stability. However, as $\varphi$ increases, the constriction factor, $K$ decreases and diversification is reduced, yielding slower response.

Typically, when the constriction factor is used, $\varphi$ is set to 4.1 (i.e. $c_1$, $c_2 = 2.05$) and the constant multiplier $K$ is thus 0.729. This results in the previous velocity being multiplied by 0.729 and the terms $\left(Pbest_i^k - X_i^k\right)$ and $\left(Gbest^k - X_i^k\right)$ being multiplied by $0.729 \times 2.05 = 1.49445$ (times a random number between 0 and 1).

The constriction factor approach results in convergence of the individuals over time. Unlike other evolutionary computation methods, the constriction factor approach ensures the convergence of the search procedure based on the mathematical theory. Therefore, the constriction factor approach can generate higher quality solutions than the basic PSO approach. However, the constriction factor approach only considers dynamic behavior of one individual and the effect of the interaction among individuals. Namely, the equations were developed with a fixed set of best positions $(Pbests\ \text{and}\ Gbest)$, although $Pbests$ and $Gbest$ change during the search procedure in the basic PSO equation.

*C. Constriction Factor Based PSO for ED Problems*

In this section, the constriction factor based PSO (CFBPSO) algorithm will be described in solving the ED problem. Details on how to deal with the equality and inequality constraints of the ED problem when modifying each individual's searching point are based on the improved PSO (IPSO) method proposed by Park (Park et al. 2006).

In subsequent sections, the detailed implementation strategies of the proposed CFBPSO method are described.

*1) Initialization of Individuals*

In the initialization process, a set of individuals (i.e. a group) is created at random within the system constraints. In this paper, an individual for the ED problem is composed of a set of elements (i.e., generator outputs). Thus, individual $i$ at iteration 0 can be represented as the vector $P_i^0 = (P_{i1}, ..., P_{in})$ where $n$ is the number of generators. The velocity of individual $i$ at iteration 0 can be represented as the vector $V_i^0 = (V_{i0}, ..., V_{in})$ and this corresponds to the generation update quantity covering all generators. The elements of position and velocity have the same dimension (i.e., MW) in this case. Note that individuals initialized must satisfy the equality constraint (3) and inequality constraints (4) defined in Section 2. That is, the sum of all elements of individual $i$ (i.e., $\sum_{j=1}^{n} P_{ij}$) should be equal to the total system demand (i.e., $P_{Demand}$) neglecting transmission losses (i.e., $P_{Loss} = 0$) and the created element $j$ of individual $i$ at random (i.e., $P_{ij}$) should be located within its boundary. Unfortunately, the created position of an individual is not always guaranteed to satisfy the inequality constraints (4). Provided that any element of an individual violates the inequality constraints then the position of the individual is fixed to its maximum/minimum operating point as follows:

$$P_{ij}^{k+1} \begin{cases} P_{ij}^{k} + V_{ij}^{k+1} & if \quad P_{ij,\min} \le P_{ij}^{k} + V_{ij}^{k+1} \le P_{ij,\max} \\ P_{ij,\min} & if \quad P_{ij}^{k} + V_{ij}^{k+1} < P_{ij,\min} \\ P_{ij,\max} & if \quad P_{ij}^{k} + V_{ij}^{k+1} > P_{ij,\max} \end{cases} \tag{11}$$

Although the previously mentioned method always produces the position of each individual satisfying the required inequality constraints (4), the problem of satisfying the equality constraint (3) still remains to be solved. Thus, it is necessary to employ a strategy suggested in the IPSO paper (Park et al. 2006) such that the summation of all elements in an individual is equal to the total system demand. The following procedure is employed for any individual in a group:

Step 1) Set $j = 1$.

Step 2) Select an element (i.e., generator) of individual $i$ at random and store in an index array $A(n)$.

Step 3) Create the value of the element (i.e., generation output) at random satisfying its inequality constraints.

Step 4) If $j = n-1$ then go to Step 5, otherwise $j = j+1$ and go to Step 2.

Step 5) The value of the last element of individual $i$ is determined by subtracting $\sum_{j=1}^{n-1} P_{ij}$ from the $Demand$. If the value is within its boundary then go to Step 8, otherwise adjust the value using (11).

Step 6) Set $l = 1$.

Step 7) Readjust the value of element $l$ in the index array $A(n)$ to the value satisfying the equality condition (i.e., $Demand - \sum_{\substack{j=1 \\ j\ne l}}^{n} P_{ij}$). If the value is within its boundary then go to Step 8; otherwise, change the value of element $l$ using (11). Set $l = l+1$, and go to Step 7. If $l = n+1$, go to Step 6.

Step 8) Stop the initialization process.

After creating the initial position of each individual, the velocity of each individual is also created at random. The following strategy is used in creating the initial velocity:

$$\left(P_{ij,\min} - \varepsilon\right) - P_{ij}^{0} \le V_{ij} \le \left(P_{ij,\max} + \varepsilon\right) - P_{ij}^{0} \tag{12}$$

where $\varepsilon$ is a small positive real number. The velocity element $j$ of individual $i$ is generated at random within the boundary.

The initial $Pbest$ of individual $i$ is set as the initial position of individual $i$ and the initial $Gbest$ is determined as the position of the individual with minimum payoff of equation (1).

2) *Updating The Velocity and Position of Individuals*

In order to modify the position of each individual, it is necessary to calculate the velocity of each individual in the next stage (i.e., generation). This can be calculated using equations (9) and (10). When the search algorithm in the CFBPSO method looks for an optimal solution in a solution space, it has a velocity multiplied by the constriction factor $K$ of equation (10) instead of $\omega$ in the basic PSO. Then velocity of each individual is restricted in the range of $[-V_{\max}, V_{\max}]$ where $V_{\max}$ is the maximum velocity. This prevents excessively large steps during the initial phases of the search.

The position of each individual is modified by equation (8). Since the resulting position of an individual is not always guaranteed to satisfy the equality and inequality constraints, the modified position of an individual is adjusted by (11). Additionally, it is necessary for the position of an individual to satisfy the equality constraint (3) at the same time. To resolve the equality constraint problem without intervening the dynamic process inherent in the PSO algorithm, the following heuristic procedures are employed:

Step 1) Set $j = 1$.

Step 2) Select an element (i.e., generator) of individual $i$ at random and store in an index array $A(n)$.

Step 3) Modify the value of element $j$ using (8), (9), and (11).

Step 4) If $j = n-1$ then go to Step 5, otherwise $j = j+1$ and go to Step 2.

Step 5) The value of the last element of individual $i$ is determined by subtracting $\sum_{j=1}^{n-1} P_{ij}$ from the $Demand$. If the value is not within its boundary then adjust the value using (11) and go to Step 6, otherwise go to Step 8.

Step 6) Set $l = 1$.

Step 7) Readjust the value of element $l$ in the index array $A(n)$ to the value satisfying the equality condition (i.e., $Demand - \sum_{\substack{j=1 \\ j\ne l}}^{n} P_{ij}$). If the value is within its boundary then go to Step 8; otherwise, change the value of element $l$ using (11). Set $l = l+1$, and go to Step 7. If $l = n+1$, go to Step 6.

Step 8) Stop the modification procedure.

The fuel cost for each individual considering the valve-point effect is calculated based on (5). The objective function of individual $i$ is obtained by summing the fuel cost for each generator in the system as shown in (1).

3) *Updating Pbest and Gbest of Individuals.*

The $Pbest$ of each individual at iteration $k+1$ is updated as follows:

$$Pbest_{ij}^{k+1} = X_{ij}^{k+1} \quad if \;\; F_i^{k+1} < F_i^k \tag{13}$$

$$Pbest_{ij}^{k+1} = Pbest_i^k \quad if \;\; F_i^{k+1} > F_i^k \tag{14}$$

where,

$F_i^{k+1}$ the objective function evaluated at the position of individual *i* at iteration *k+1*

$F_i^k$ the objective function evaluated at the position of individual *i* at iteration *k*

$X_{ij}^{k+1}$ position of individual $i$ at iteration $k+1$

$Pbest_{ij}^{k+1}$ best position of individual $i$ until iteration $k+1$

(13) and (14) compare the $Pbest$ of every individual with its current fitness value. If the new position of an individual has better performance than the current $Pbest$, the $Pbest$ is replaced by the new position. In contrast, if the new position of an individual has lower performance than the current $Pbest$, the $Pbest$ value remains unchanged.

Additionally, the $Gbest_{ij}^{k+1}$ global best position at iteration $k+1$ is set as the best evaluated position among $Pbest_{ij}^{k+1}s$. In other words, $Gbest$ determines the current best fitness value in the entire population. If the current best fitness value among all other $Pbests$ is better than the current $Gbest$, then the position of the current best fitness vale is assigned to $Gbest$.

4) *The Stopping Criteria*

The proposed CFBPSO is terminated if the iteration approaches a predefined criteria, usually a sufficiently good fitness or in this case, a predefined maximum number of iterations (generations).

## 4. CASE STUDIES

In order to verify the feasibility of the proposed CFBPSO method, and make a comparison with the improved PSO (IPSO) method researched by Park (Park et al. 2006), a three-generator power system was tested. In both cases, the test system is composed of three generating units and the input data of the 3-generator system are as given in Table I. The valve-point effects are considered and the transmission loss is omitted. The total demand for the system is set to 850MW.

Table I: Data For Test Case (3-Unit System)

| Unit | $a_i$ | $b_i$ | $c_i$ | $e_i$ | $f_i$ | $P_{i,\min}$ | $P_{i,\max}$ |
|---|---|---|---|---|---|---|---|
| 1 | 561 | 7.92 | 0.001562 | 300 | 0.0315 | 100 | 600 |
| 2 | 310 | 7.85 | 0.001940 | 200 | 0.0420 | 100 | 400 |
| 3 | 78 | 7.97 | 0.004820 | 150 | 0.0630 | 50 | 200 |

### A. *Case I*

In this case, the fuel cost characteristics considering the valve-point effects were employed to test and verify the feasibility of the proposed CFBPSO. In order to simulate the proposed CFBPSO method, some parameters must be assigned and are as follows:

- Number of particles = 50;
- Maximum iteration number = 10000;
- The convergence rate of the system is controlled by $\varphi$. In this case, $\varphi$ is set to 4.1 (i.e. $c_1$, $c_2 = 2.05$) and the constriction factor $K$ is thus 0.729.

The obtained results for the three-generator system using the CFBPSO method are given in Table II and the results are compared with those from GA (Walters and Sheble 1993), EP (Yang et al. 1996), MPSO (Hou et al. 2005), and IPSO (Park et al. 2006). As shown in Table II, the CFBPSO has outperformed GA and has provided the same optimal solution as obtained by EP, MPSO and IPSO.

Table II: Comparison of Simulation Results of Each Method Considering Valve-Point Effect (3-Unit System)

| Unit | GA | EP | MPSO | IPSO | CFBPSO |
|---|---|---|---|---|---|
| **1** | 300.00 | 300.26 | 300.27 | 300.27 | 300.27 |
| **2** | 400.00 | 400.00 | 400.00 | 400.00 | 400.00 |
| **3** | 150.00 | 149.74 | 149.73 | 149.73 | 149.73 |
| **TP** | 850.00 | 850.00 | 850.00 | 850.00 | 850.00 |
| **TC** | 8237.60 | 8234.07 | 8234.07 | 8234.07 | 8234.07 |

* TP: Total Power [MW], TC: Total Generation Cost [$]

Table III shows the frequency of attaining the best cost of 8234.07 [$] out of 50 runs for two algorithms of IPSO and CFBPSO with 50 particles and 10,000 generations. In this table, IPSO represents the improved PSO algorithm by Park (Park et al. 2006) and CFBPSO is the algorithm presented in this paper.

Table III: Relative Frequency of Convergence (50 Runs)

| | Best Cost = 8234.07 [$] | Probability |
|---|---|---|
| **IPSO** | 7 | 0.14 |
| **CFBPSO** | 14 | 0.28 |

The results in Table III illustrated that CFBPSO has higher probability of achieving the better solutions between both algorithms.

### B. *Case II*

In this case, the maximum number of generations is set to 100 and the number of particles limited to 5. In order to further test the performance of the proposed CFBPSO method when compared to the IPSO in solving the non-smooth ED problem with valve-point effects, both methods were applied on a three-generator system where the fitness of the best particle for each method was being investigated.

The criterion for the comparison is the achievement of the best cost of 8234.07 [$] in the shortest generation. In addition to that, observations were made on the best, mean, and worst costs, and standard deviation. Fig. 3 shows the fitness value of the best particle for both methods and Table IV summarizes the results of both simulations.

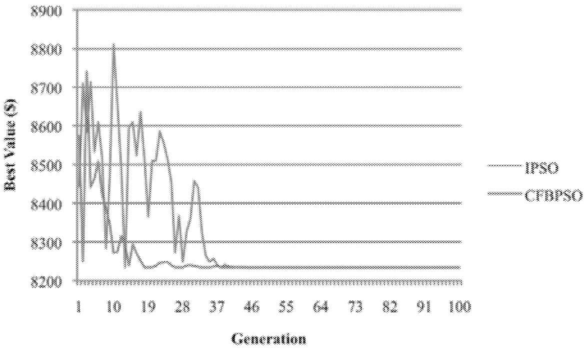


Fig. 3: Fitness of the best particles for the IPSO and CFBPSO.

Table IV: Detailed Comparison of The IPSO and CFBPSO (3-Unit System, 5 Particles, and 100 Generations)

| | Cost [$] | | | STD | BCG |
|---|---|---|---|---|---|
| | Best | Mean | Worst | | |
| **IPSO** | 8234.07 | 8319.90 | 8810.15 | 145.42 | 47 |
| **CFBPSO** | 8234.07 | 8258.45 | 8739.77 | 76.12 | 45 |

* STD: STANDARD DEVIATION, BCG: BEST COST GENERATION NO.

The simulation results indicate that the CFBPSO method exhibits good performance. As seen in Table IV, the CFBPSO finds the cheapest cost faster than the IPSO. From Fig. 3, it can be observed that the IPSO suffers from a lot of fluctuations in reaching the optimal result in different generations. In this respect the superiority of the CFBPSO is quite evident.

## 5. CONCLUSION

This paper presents a novel approach for solving the non-smooth ED problem with valve-point effects based on the constriction factor based PSO (CFBPSO). The proposed CFBPSO includes a constriction factor $K$ into the velocity rule, which has an effect of reducing the velocity of the particles as the search progresses thereby ensures convergence of the particle swarm algorithm. The appropriate parameter values for the ED by the constriction factor approach are the same as those recommended by other PSO papers. The robust convergence characteristic of the CFBPSO method is also ensured in solving the ED problem. Simulation results on the three-generator system demonstrate the feasibility and effectiveness of the CFBPSO method in minimizing cost of the generation. The results also show that improvements were made to solve the ED problem more effectively. Finally, it has been demonstrated that the CFBPSO algorithm improves the convergence and performs better when compared with the improved PSO (IPSO).

## REFERENCES

Clerc M. 1999. “The swarm and the queen: towards a deterministic and adaptive particle swarm optimization,” *Proceedings of the 1999 Congress on Evolutionary Computation*, Vol. 3, pp. 1951-1957.

Clerc M. and J. Kennedy. 2002. “The particle swarm-explosion, stability, and convergence in a multidimensional complex space,” *IEEE Trans. on Evolutionary Computation*, Vol. 6, No. 1, pp. 58-73.

Eberhart R.C. and Y. Shi. 2000. “Comparing inertia weights and constriction factors in particle swarm optimization”, *Proceedings of the 2000 Congress on Evolutionary Computation*, Vol. 1, pp. 84-88.

Gaing Z.L. 2003. “Particle swarm optimization to solving the economic dispatch considering the generator constraints”, *IEEE Trans. on Power Systems*, Vol. 18, No. 3, pp. 1187-1195.

Hou Y.H.; L.J. Lu; X.Y. Xiong; and Y.W. Wu. 2005. “Economic dispatch of power systems based on the modified particle swarm optimization algorithm”, *Transmission and Distribution Conference and Exhibition: Asia and Pacific, IEEE/PES*, pp. 1-6.

Kennedy J. and R.C. Eberhart. 1995. “Particle swarm optimization”, *Proceedings of IEEE International Conference on Neural Networks (ICNN’95)*, Vol. IV, pp. 1942-1948, Perth, Australia.

Kennedy J. and R.C. Eberhart. 2001. *Swarm Intelligence*. San Francisco, CA: Morgan Kaufmann.

Lee K.Y.,; A. Sode-Yome; and J.H. Park. 1998. “Adaptive Hopfield neural network for economic load dispatch”, *IEEE Trans. on Power Systems*, Vol. 13, No. 2, pp. 519-526.

Liang Z.X. and J.D. Glover. 1992. “A zoom feature for a dynamic programming solution to economic dispatch including transmission losses”, *IEEE Trans. on Power Systems*, Vol. 7, No. 2, pp. 544-550.

Lin W.M.; F.S. Cheng; and M.T. Tsay. 2002. “An improved Tabu search for economic dispatch with multiple minima”, *IEEE Trans. on Power Systems*, Vol. 17, No. 1, pp.108-112.

Park J.B.; K.S. Lee; J.R. Shin; and K.Y. Lee. 1993. “A particle swarm optimization for economic dispatch with nonsmooth cost functions”, *IEEE Trans. on Power Systems*, Vol. 8, No. 3, pp. 1325-1332.

Shi Y. and R.C. Eberhart. 2001. “Particle swarm optimization: developments, applications, and resources”, *Proceedings of the 2001 Congress on Evolutionary Computation*, Vol. 1, pp. 81-86.

Walters D.C. and G.B. Sheble. 1993. “Genetic algorithm solution of economic dispatch with the valve-point loading”, *IEEE Trans. on Power Systems*, Vol. 8, No. 3, pp. 1325-1332.

Wood A.J. and B.F. Wollenberg. 1996. *Power Generation, Operation and Control, 2nd Edition*, New York: John Wiley & Sons.

Yang H.T.; P.C. Yang; and C.L. Huang. 1996. “Evolutionary programming based economic dispatch with the valve point loading”, *IEEE Trans. on Power Systems*, Vol. 11, No. 1, pp. 112-118.

# STATISTICAL MODELS FOR PRICING WEATHER DERIVATIVES FOR SOUTH AFRICAN COASTAL AREAS

Mark W. Nasila
Igor N. Litvine
Department of Statistics
Nelson Mandela Metropolitan University
PO Box 77000
Port Elizabeth
E-mail: Mark.Nasila@nmmu.ac.za and Igor.Litvine@nmmu.ac.za

**KEYWORDS**

Coastal areas, Weather derivatives, Weather risks,

## ABSTRACT

Coastal areas are always an ideal place for different specific business activities which include the agricultural industry. This study reviews available statistical models for pricing weather derivatives with temperature as the underlying variable which will enable industries, businesses and other organizations in South African coastal areas like Durban, Cape Town and Port Elizabeth to protect themselves against losses due to fluctuation in the weather and thus hedge their risks. Historical South African weather data was used to test the models.

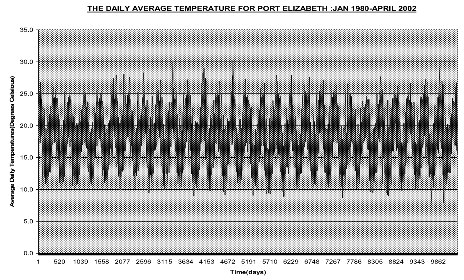
## 1. INTRODUCTION

Weather derivatives are financial contracts with payouts that depend on weather in some form. West (2002) defines a weather derivative as a contract that provides a payoff in response to an index level based on weather phenomena. The underlying variable can be, for example; humidity, rain, snowfall, temperature, or even sunshine. The main players who take part in the weather derivatives markets industry can be grouped in to five main categories, namely: End users (also referred to as hedgers), Speculators, Market makers, Brokers and Insurance and re-insurance companies.

## 2. METHODOLOGY AND RESULTS

Since it is essential for anyone playing a role in weather derivatives to understand the behavior of temperature, this section investigates a model that describes the daily temperature. Figure 3.1.1 below graphs the real Port Elizabeth temperature data for 28 years (Source of data: South African Weather Service).

Figure 3.1.1: A 28-year segment of the daily average temperature for Port Elizabeth: Jan 1980 - April 2008



From the daily average temperature data in figure 3.1.1 one clearly sees that there is cyclical seasonal variation in the temperature. The daily mean temperature seems to vary between approximately $26^{0}$C during the summer periods and approximately $12^{0}$C during the winter periods. When one looks at figure 3.1.1, it is possible to model the seasonal dependence with some function, say a sine function (Alaton, 2003). The sine function would have the following form:

$$A + C\sin(\omega t + \varphi) \quad \text{……….} (3.1.1)$$

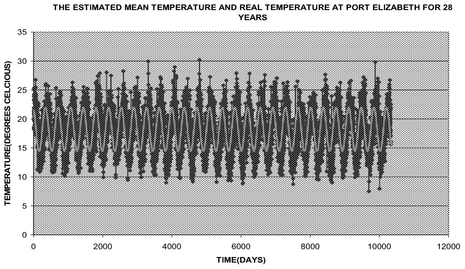
Where $t$ denotes the time measured in days, $\omega$ denotes the period and $\varphi$ denotes the phase angle. Upon closer inspection of the data one may observe that it reveals a positive trend. This trend might be difficult to determine as it is weak, however, it does exist. The daily average temperature tends to increase over the years. There are many factors which contribute towards the weak positive trend. It might be due to the global warming effect which is causing a warming trend to be being experienced on our planet. Also, there is the urban heating effect, temperatures tend to increase in areas nearby a big city because the city is growing and warming its surroundings with the heat generated from the industries within the city.

One may make the assumption that this warming trend is linear due to the fact that the trend is weak (Alaton, 2003). With all these characteristics, one combines the sine and linear functions to end up with a model for the mean temperature at time $t$, which has the form:

$$T^{m}_{t} = A + Bt + C\sin(\omega t + \varphi) \text{ ......... (3.1.2)}$$

The parameters $A, B, C, \varphi$ and $\omega$ have to be determined in order to fit the above curve to the real temperature data. Fitting the above function (3.1.2) to the temperature data using the method of least squares, results in the determination of the numerical estimates of the parameters $A, B, C, \varphi$ and $\omega$ . Applying this method to the 10336 observations of the temperature data from Port Elizabeth, the amplitude of the sine function is about $3.80^{0}$C which implies that the temperature difference between a typical winter day and a summer day is about $7.6^{0}$C. A plot of this function together with the real temperature data in shown in figure 3.1.2 below. The red smooth curve represents the fitted model while the dark blue one represents the real data.

Figure 3.1.2: The estimated mean temperature and real temperature at Port Elizabeth for 28 years



## 3. PRICING WEATHER DERIVATIVES

The process of pricing weather derivatives is an extensive and involving process because the two parties try to hedge the risk created by the weather, which has no value in the market as it is not a tradable asset. The first stage involves pricing the expected payoff at the end of a contract while second one requires one to estimate a fair price which one should pay for their risk to be hedged; this involves including the market price of the underlying risk.

### Expected Derivative Pay-Off

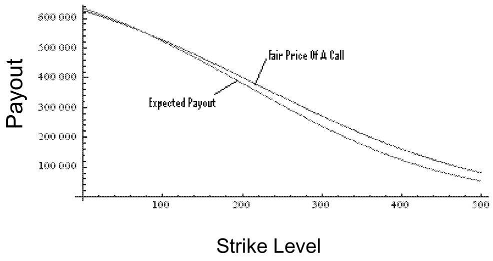
It was decided to combine the distribution chosen for pricing our derivatives and the commonly used structures (Jewson, 2005). In this case the normal distribution is the more favourable distribution. Only the call contract will be discussed in this paper.

A call option is a financial contract between two parties where the buyer of a call option has the right, but not the obligation to buy an agreed quantity of a particular financial asset or commodity from the seller of the option at a particular time. In this case, the buyer pays a premium for this right. The seller is also obliged to sell the commodity should the buyer decide to purchase. In weather derivatives, at the beginning of the contract the buyer pays a premium to the seller and at the end of the contract the seller pays the buyer a pay-off which depends on the value of the index accumulated

### Fair Price of a Weather Derivative

The fair price of a derivative is the cost of a derivative including the market price of a risk. The market price of a risk is normally included in the calculation of the expected temperature. If a standardised market risk of 0.08 is included, it is found that the same call contract costs more. Figure 4.1.2 below shows the relationship between the expected payoff and the fair price of a contract with increase in strike level.

Figure 4.1.2: Graphs comparing the expected payout and fair price of a call contract over different strike levels.



It is clear that an increase in the strike level from approximately eighty degree days increases the difference between the expected payout and the fair price, which includes the market risk.

## 4. CONCLUSION

This study justifies the need for a standard pricing approach for industries affected by extreme weather conditions so that all participants who wish to hedge against extreme weather could start communicating in a common language. It enables many companies and business organisations establish a hedging policy or even figure out how their businesses or industries are exposed to weather risks.

## LIST OF REFERENCES

Alaton, P., Djehiche, B. and Stillberger, D. (2003), "On Modelling and Pricing Weather Derivatives", Applied Mathematical Finance, Volume 9, Issue 1.

Jewson, S., Brix, A. and Ziehmann, C. (2005), "Weather Derivative Valuation", Cambridge Univ. Press, New York.

South African Weather Service (SAWS). Temperature Data: 1980-2008. SAWS Weather results and Press release. Online:. http://www.weathersa.co.za/Climat/Climstats/PortElizabethStats.jsp

West, J. (2002), "Benchmark Pricing of Weather Derivatives*". Working Paper, School of Finance and Economics, University of Technology, Sydney.

Personal Communication: Micali, V. (2008). Technical Director, Eskom. Personal interview at Nelson Mandela Metropolitan University, Port Elizabeth. May, 2008

## BIOGRAPHY

**Mark Nasila** was born in Trans-Nzoia, Kenya. In 2003 he was admitted to Walter Sisulu University (South Africa) where he studied Statistics and Computer Science. In 2007 he was admitted to Nelson Mandela Metropolitan University (RSA) where he obtained his BSc Honours in Mathematical Statistics. He obtained his MSc degree (with distinction) in 2008 and is currently a full time PhD student in the same University.

**IGOR LITVINE** was born in Kiev, Ukraine. In 1975 he was admitted to the Kiev State (Shevchenko) University where he studied Statistics, Applied Mathematics and Computer Science. He obtained his MSc degree (with distinction) in 1980 and PhD in 1984. From 1983 till 1989 he was lecturing in the same University. From 1989 he has been working in Africa: Ethiopia, Lesotho and RSA (since 1995). Nowadays Prof Litvine is Head of Statistics department in the Nelson Mandela Metropolitan University (Port Elizabeth, South Africa).

# MODELLING OF INTERVENTION EFFECT ON TRUST

Arnostka Netrvalova
Jiri Safarik
Department of Computer Science and Engineering, Faculty of Applied Sciences
University of West Bohemia
30614 Plzen, Univerzitni 8, Czech Republic
E-mail: netrvalo@kiv.zcu.cz

**KEYWORDS**
Trust, interpersonal trust, trust modelling, trust intervention.

## ABSTRACT

The paper deals with interpersonal trust modelling taking the focus on intervention effect. Terms as trust, trust factors and trust representation are introduced. Brief description of interpersonal trust forming is presented. Intervention model is introduced. The proposed trust formula formation and intervention model enable to cover trust intervention. The study of the behaviour of trust evolution is extended in this way. The comparison of trust model with and without intervention effect is discussed. The parameter values of trust intervention, i.e. intervention power, and intervention distribution are modified and the effect of these modifications on trust is studied.

## INTRODUCTION

Trust is a unique phenomenon and plays an important role in the relationships among subjects in the communities. These subjects need not be only humans. In the internet age, the trust among humans and the machines, e.g. servers, network nodes or various software utilities, gains more and more on importance. Widening of e-service (Liu et al., 2008), e-commerce (Zhang et al., 2008), e-banking, etc., arises the question of human machine trust. Further, trust plays an important role in peer-to-peer networks (Wu et al., 2008), ad hoc networks (Mejia et al. 2009), grid computing, semantic web (Wang and Zhang, 2008), and multi agent systems, where humans and/or machines have to collaborate. Trust models are used in those uncertain environments (Wang and Varadharajan, 2008), (Camp et al., 2008), (Velloso et al., 2006). The role of trust is very important in e-service, e-banking and e-commerce particularly, e.g. (Chen and Yeager, 2008).

The acceptance of trust is wide and various explanations are offered (Fetzer, 1988); from honesty, truthfulness, confident expectation or hope, something managed for the benefit of another, confidence in ability or intention to pay for goods or services in the future, till business credit. The universal trust definition does not exist. Bulk of definitions comes out from Gambetta's definition (Gambetta, 2000). We will understand trust as a given credit, hope, confidence in ability or intention of some subject to perform to benefit of other subject at some future time.

## TRUST FORMING FACTORS

Trust models, and interpersonal trust models particularly, e.g. (Wu et al., 2008), (Lifen, 2008), (Ryutov et al., 2007), (Chen and Yeager, 2008) are usually focused on one of factors determining trust, but no more than two were considered. The reputation, recommendation and initial trust are basic factors determining trust. Initial trust is the trust value in other person on the start. The reputation represents the knowledge about trusted person. The information obtained by communication with others is called recommendation. Each of the factors (initial trust, reputation, and recommendations) can be modelled as an individual component.

Firstly, for examining trust as a behavioural pattern, some way of representing trust is needed. Generally, trust can be quantified by a value from the interval $\langle a, b\rangle$, where $a$, $b$ ($a < b$) are integer or real numbers. Value $a$ represents complete distrust and value $b$ means blind trust. Without loss of generality, we will use real values from the interval $\langle 0, 1\rangle$.

Next, we specify an interpersonal trust representation, i.e. trust between two subjects. Consider a group of $n$ subjects represented as the set $S = \{s_1, s_2, \ldots, s_n\}$. The measure of interpersonal trust between the subject $s_i$ and $s_j$ is introduced by:

$$t_{ij} = t(s_i, s_j),\ t_{ij} \in \langle 0, 1\rangle, \tag{1}$$

where: $i, j = 1, \ldots, n$ and $i \neq j$.

We use a matrix, called interpersonal trust matrix, for representation of interpersonal trust in a group, where $t_{ij}$ are matrix entries. Matrix entry -1 denotes that the subject does not know this one or self-trust (self-trust is not considered).

## INTERVENTION EFFECT MODEL

We will use a general model of information intervention effect depicted in Figure 1 (Vavra F., University of West Bohemia, personal communication).

The probability distribution $P$ on the input represents the state before intervention, the probability distribution $Q$ on the output describes the state after intervention activity and the intervention is modelled by probability distribution $R$, where $x$ ($x \in X$) is event observed from finite set of events $X$ . These events can be products preference and the probability distribution represents their relative sale frequencies.

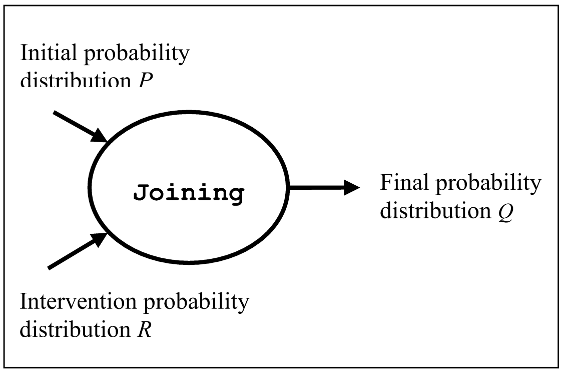

Initial probability distribution $P$
Joining
Final probability distribution $Q$
Intervention probability distribution $R$

Figure 1: Model of Information Intervention Effect.

The uncertainty of preference in a single observation can be measured by entropy:

$$H(X) = -\sum_{x \in X} P(x) \cdot \lg P(x) \tag{2}$$

where $P(x)$ is the value of probability mass function. The entropy is increasing if the effect of the intervention uniforms probability distribution, whereas it is decreasing in opposite case.

The difference between initial probability distribution $P$ and final probability distribution $Q$ can be measured by relative entropy:

$$D(P \| Q) = \sum_{x \in X} P(x) \lg \frac{P(x)}{Q(x)} \tag{3}$$

While relative entropy is not a metric, we will take symmetric relative entropy:

$$d(p,q) = D(P \| Q) + D(Q \| P) \tag{4}$$

For joining initial probability and intervention probability we use their mixture (Rényi, 1961):

$$Q(x) = (1-\lambda)P(x) + \lambda\, R(x), \tag{5}$$

where $0 < \lambda \le 1$, represents intensity of the intervention.

Given probability mass functions $P(x)$, $R(x)$, $Q(x)$ the intensity $\lambda$ can be found by

$$\lambda = \frac{\sum_{x \in X} (Q(x) - P(x))(R(x) - P(x))}{\sum_{x \in X} (R(x) - P(x))^2}, \tag{6}$$

if exists.

The conditions of existence are

$$\sum_{x \in X} (R(x) - P(x))(Q(x) - R(x)) \le 0$$

and

$$\sum_{x \in X} (R(x) - P(x))(Q(x) - P(x)) \ge 0 \cdot$$

The mixture of distributions may be used for negative $\lambda$ if following holds:

$$0 \le (1-\lambda)p(x) + \lambda\, r(x) \le 1;\ \forall x \in X. \tag{7}$$

## TRUST EVOLUTION MODEL

Trust between subjects evolves under changing factors determining trust. We have proposed trust model determining new value of interpersonal trust $T_{ij}$ of subject $s_i$ to subject $s_j$ as function of trust forming factors, i.e. previous trust each to other, subject reputation, number of subject recommendations, number of reciprocal contacts and trusting disposition (Netrvalova and Safarik, 2009). Initial trust between subjects is got on the start. The reputation of the subject comes after individual experience and by some information dissemination about subject in its neighbourhood and influences trust formation considerable. Trust depends also on the frequency of mutual contacts of subjects. Next, trust is formed by information about another subject that other subjects have passed on. This information is called recommendation. Trusting disposition representing a degree of non rational behaviour of a subject is modelled by a random factor.

Trust forming of $i$-th subject (trustor) to $j$-th subject (trustee) is described by

$$T_{ij} = t_{ij} + \sqrt{t_{ij}\, t_{ji}} \left( \frac{\Delta c_{ij}}{w_{ci}} + \frac{\Delta d_{ij}}{w_{di}} \right) \frac{r_{ij}}{w_{ri}} \frac{g_{ij(\alpha,\beta)}}{w_{g_i}} \tag{8}$$

where $T_{ij}$ ($0 \le T_{ij} \le 1$) is new trust value of $i$-th subject in $j$-th one, $t_{ij}$ is previous trust (trust starting value is $t_{0ij}$) of $i$-th subject in $j$-th one, $t_{ji}$ is previous trust of $j$-th subject in $i$-th one, $\Delta c_{ij}$ is relative gain (loss) of the number of contacts between $i$-th and $j$-th subject, $\Delta d_{ij}$ is relative gain (loss) of the number of recommendations of $j$-th subject to $i$-th subject, $r_{ij}$ is reputation of $i$-th subject about $j$-th one, $g_{ij(\alpha,\beta)}$, $0<\alpha<\beta\le 1$ is trusting disposition probability distribution, $w_{ci}$ is weight coefficient of the number of contacts of $i$-th subject, $w_{di}$ is weight coefficient of the number of recommendation of $j$-th subject to $i$-th subject, $w_{ri}$ is weight coefficient of effect of reputation of $i$-th subject about $j$-th one, and $w_{gi}$ is weight coefficient of trusting disposition.

The model reflects usual factors influencing interpersonal trust in standard real life situations. On the other hand, there are situation when there is a massive intervention in order to increase trust to some subject(s), e.g. election campaign.

To model trust intervention, we use described intervention effect model.

Initial probability distribution is given by initial trust values $T_{ij}$ of a subject $s_i$ to all other subjects. The intention to increase trust value to some subject is described by intervention probability distribution $I$. Expressing the intensity of intervention by$\lambda$, $0 < \lambda \le 1$, the new trust probability distribution is given by values $T'_{ij}$

$$T'_{ij} = (1- \lambda)\, T_{ij} + \lambda\, I_{ij}, \tag{9}$$

## EXPERIMENTS AND RESULTS

To pursue trust intervention model behaviour we carried out series of experiments. Here, we present following scenario. The group of five subjects was considered. Relations of subject $s_1$ to other subjects, i.e. $s_2$, $s_3$, $s_4$, and $s_5$, from this group were chosen. Parameters of experiments are presented in Table 1 (initial trust), Table 2 (reputation), Table 3 (number of mutual contacts), Table 4 (number of recommendation), Table 5 (trusting disposition) and Table 6 (intervention distribution).

Table 1: Initial Trust of Subject $s_1$ to Other Subjects

| $s_1\to s_2$ | $s_1\to s_3$ | $s_1\to s_4$ | $s_1\to s_5$ |
|---|---|---|---|
| 0,75 | 0,04 | 0,2 | 0,01 |

Table 2: Subject $s_1$ - Partner's Reputation of Other Subjects

| $r_2\to s1$ | $r_3\to s_1$ | $r_4\to s_1$ | $r_5\to s_1$ |
|---|---|---|---|
| 0,25 | 0,23 | 0,3 | 0,14 |

Table 3: Number of Contacts of Subject $s_1$ with Partners

| Step | $s_1\to s_2$ | $s_1\to s_3$ | $s_1\to s_4$ | $s_1\to s_5$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 2 | 1 | 0 | 1 | 0 |
| 3 | 1 | 0 | 2 | 0 |
| 4 | 0 | 0 | 2 | 0 |
| 5 | 0 | 0 | 3 | 0 |

Table 4: Partner's Recommendation to Subject $s_1$

| Step | $d_2\to s_1$ | $d_3\to s_1$ | $d_4\to s_1$ | $d_5\to s_1$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 2 | 1 | 0 | 2 | 0 |
| 3 | 0 | 0 | 3 | 0 |
| 4 | 0 | 0 | 2 | 0 |
| 5 | 0 | 1 | 4 | 3 |

Table 5: Trusting Disposition (Subject $s_1$ to Others) Stepwise

| Step | $s_1\to s_2$ | $s_1\to s_3$ | $s_1\to s_4$ | $s_1\to s_5$ |
|---|---|---|---|---|
| 0 | 0,50 | 0,50 | 0,50 | 0,50 |
| 1 | 0,55 | 0,88 | 0,85 | 0,34 |
| 2 | 0,74 | 0,92 | 0,67 | 0,65 |
| 3 | 0,82 | 0,62 | 0,67 | 0,65 |
| 4 | 0,71 | 0,56 | 0,76 | 0,35 |
| 5 | 0,76 | 0,78 | 0,56 | 0,91 |

Trust formation of subject $s_1$ in other subjects without intervention effect is presented in Figure 2. Trust changes were relatively small adequately to reputation values, number of contacts and recommendations. This behaviour enables us to observe changes in trust evolution caused by intervention. Intervention distribution values used in the experiment are in Table 6.

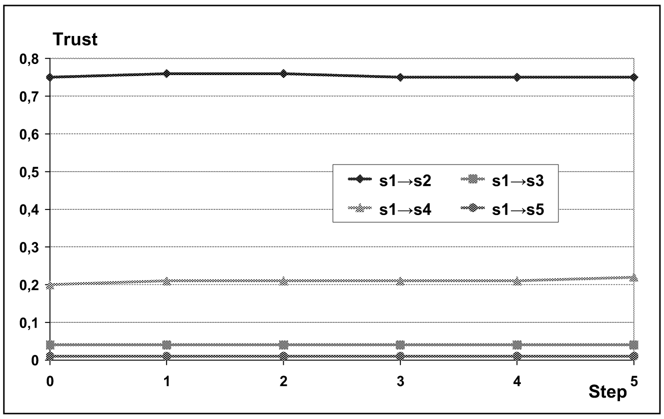


Figure 2: Trust of Subjects without Intervention Effect.

Table 6: Intervention Distribution (Subject $s_1$)

| $s_1\to s_2$ | $s_1\to s_3$ | $s_1\to s_4$ | $s_1\to s_5$ |
|---|---|---|---|
| 0,03 | 0,10 | 0,75 | 0,12 |

Values of $\lambda$ parameter were set to 0,05; 0,1; 0,5 successively. Subject $s_1$ and its partners were observed.

Stepwise trust forming is shown in Figure 3, Figure 5 and Figure 7. The influence of the intervention intensity represented by the parameter $\lambda$ can be visually observed. Moreover, the intervention effect can be measured using the terms of information theory, that were introduced previously, i.e. entropy (2), relative entropy (3) and symmetric relative entropy (4). Trust intervention effect, i.e. initial trust distribution before intervention ($T_0$), final trust distribution after first step ($T_1$), and trust intervention distribution, are depicted in Figure 4, Figure 6, and Figure 8. Entropy relative entropy and symmetric relative entropy values are presented in Table 7, Table 8 and Table 9.

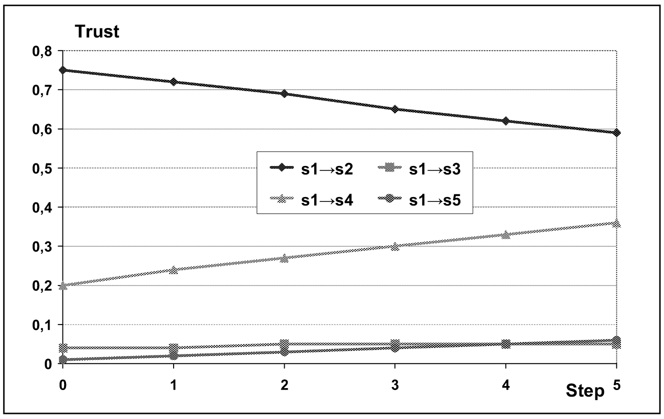


Figure 3: Trust Formation for $\lambda$ =0,05 Stepwise.

Table 7: Entropy and Relative Entropy for $\lambda$ =0,05 after the 1st Step

| Subject | $H(T_0)$ | $H(I)$ | $H(T_1)$ | $D(T_0\|\|T_1)$ | $D(T_1\|\|T_0)$ |
|---|---|---|---|---|---|
| $s_1\to s_2$ | 0,311 | 0,151 | 0,341 | 0,044 | -0,042 |
| $s_1\to s_3$ | 0,186 | 0,332 | 0,185 | 0,000 | 0,000 |
| $s_1\to s_4$ | 0,464 | 0,311 | 0,494 | -0,052 | 0,063 |
| $s_1\to s_5$ | 0,066 | 0,367 | 0,112 | -0,010 | 0,020 |
| Sum | 1,027 | 1,161 | 1,132 | -0,018 | 0,041 |
| $\lambda comp$= | 0,053 | | | $d(p, r)$ = | 0,023 |

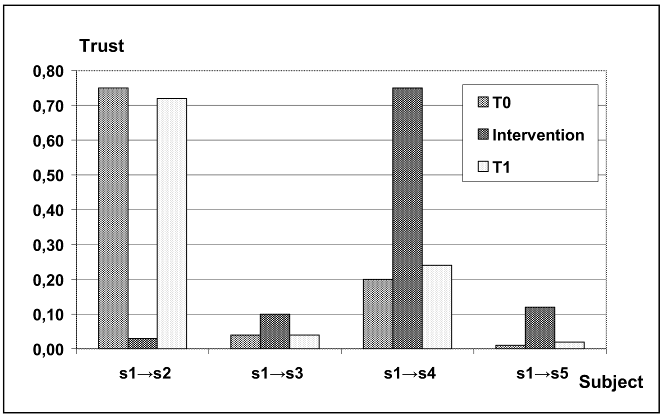


Figure 4: Trust Intervention Effect for $\lambda$ =0,05 after the First Step.

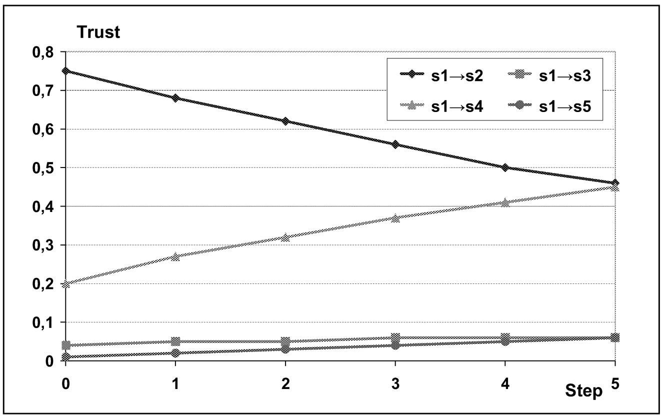


Figure 5: Trust Formation for $\lambda$ =0,1 Stepwise.

Table 8: Entropy and Relative Entropy for $\lambda$ =0,1 after the 1st Step

| Subject | $H(T_0)$ | $H(I)$ | $H(T_1)$ | $D(T_0\|\|T_1)$ | $D(T_1\|\|T_0)$ |
|---|---|---|---|---|---|
| $s_1 \rightarrow s_2$ | 0,311 | 0,151 | 0,378 | 0,106 | -0,096 |
| $s_1 \rightarrow s_3$ | 0,186 | 0,332 | 0,216 | -0,013 | 0,016 |
| $s_1 \rightarrow s_4$ | 0,464 | 0,311 | 0,510 | -0,087 | 0,117 |
| $s_1 \rightarrow s_5$ | 0,066 | 0,367 | 0,113 | -0,010 | 0,020 |
| Sum | 1,027 | 1,161 | 1,217 | -0,004 | 0,057 |
| $\lambda comp$= | 0,108 | | | $d(p, r)$ = | 0,053 |

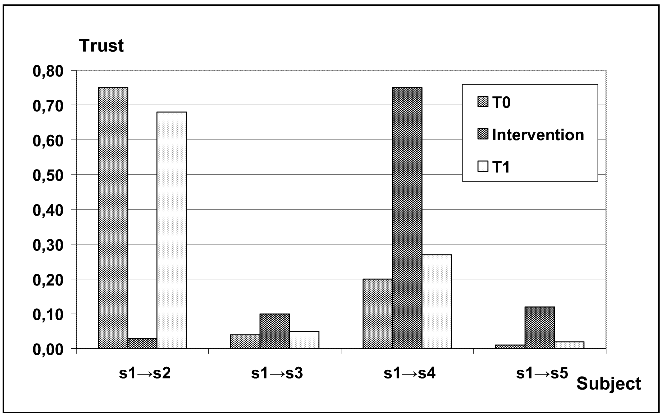


Figure 6: Trust Intervention Effect for $\lambda$=0,1 after the First Step.

The initial trust of subject $s_1$ is highest to subject $s_2$ (Table 1). The intervention is in favour of subject $s_4$ (Table 6). Clearly, the trust distribution has to become more even. This is proved by higher entropy $H(T_1)$ of new trust distribution (Table 7, Table 8, and Table 9) and the entropy increases when the intervention is stronger. Also, the new trust distribution has to have greater distance to the initial one, what is proved by higher symmetric relative entropy (Table 7, Table 8, and Table 9) and the distance grows when the intervention is stronger.

Achieving a trust distribution in some steps, we can ask on the intervention intensity $\lambda$ which would cause the same trust distribution in one step. This value is given by Equation (6).

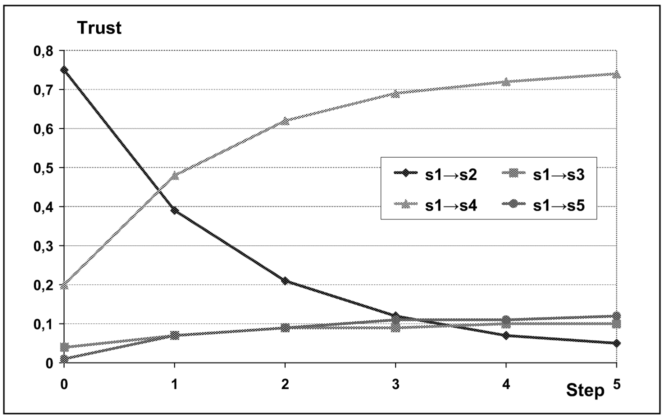


Figure 7: Trust Formation for $\lambda$ =0,5 Stepwise.

Table 9: Entropy and Relative Entropy for $\lambda$ =0,5 after the 1st Step

| Subject | $H(T_0)$ | $H(I)$ | $H(T_1)$ | $D(T_0\|\|T_1)$ | $D(T_1\|\|T_0)$ |
|---|---|---|---|---|---|
| $s_1 \rightarrow s_2$ | 0,311 | 0,151 | 0,530 | 0,708 | -0,368 |
| $s_1 \rightarrow s_3$ | 0,186 | 0,332 | 0,269 | -0,032 | 0,057 |
| $s_1 \rightarrow s_4$ | 0,464 | 0,311 | 0,508 | -0,253 | 0,606 |
| $s_1 \rightarrow s_5$ | 0,066 | 0,367 | 0,267 | -0,028 | 0,197 |
| Sum | 1,027 | 1,161 | 1,574 | 0,395 | 0,492 |
| $\lambda comp$= | 0,503 | | | $d(p, r)$ = | 0,887 |

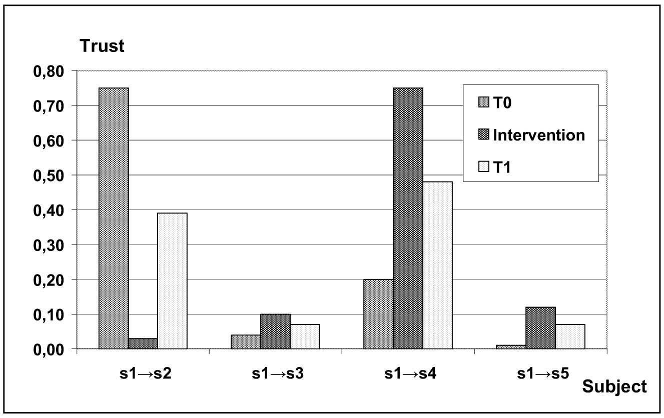


Figure 8: Trust Intervention Effect for $\lambda$ =0,5 after the First Step.

Trust values achieved for $\lambda$=0,5 after five steps would be reached by $\lambda_{comp}$=0,976 in one step. Entropy relative entropy and symmetric relative entropy values are displayed in Table 10. We can observe expected decrease of the entropy $H(T_1)$ for $\lambda_{comp}$=0,976 compared to $H(T_1)$ for $\lambda$=0,5 as the distribution values became more uneven, now on benefit of subject $s_4$. The enormous increase of symmetric relative entropy indicates a grandiose intervention power.

Table 10: Entropy and Relative Entropy for $\lambda comp$ =0,976

| Subject | $H(T_0)$ | $H(I)$ | $H(T_1)$ | $D(T_0\|\|T_1)$ | $D(T_1\|\|T_0)$ |
|---|---|---|---|---|---|
| $s_1 \rightarrow s_2$ | 0,311 | 0,151 | 0,216 | 2,930 | -0,195 |
| $s_1 \rightarrow s_3$ | 0,186 | 0,332 | 0,332 | -0,052 | 0,132 |
| $s_1 \rightarrow s_4$ | 0,464 | 0,311 | 0,321 | -0,377 | 1,397 |
| $s_1 \rightarrow s_5$ | 0,066 | 0,367 | 0,367 | -0,035 | 0,430 |
| Sum | 1,027 | 1,161 | 1,236 | 2,464 | 1,764 |
| $\lambda comp$= | 0,976 | | | $d(p, r)$ = | 4,228 |

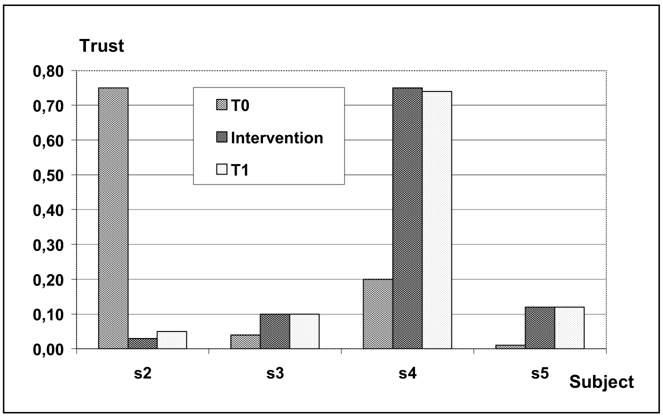


Figure 9: Trust Intervention Effect for $\lambda$ =0,976.

## CONCLUSION

We developed trust model integrating intervention effect for trust evolution. The experiments proved behaviour of the model to be in accordance with expectations.

Next, we intend to pursue the collaboration with sociologist to apply the model to real cases. The model itself will be deployed in an agent based trust management model.

## ACKNOWLEDGEMENTS

The work was granted by the Ministry of Education, Youth and Sport of the Czech Republic - University spec. research - 1311.

## REFERENCES

Camp L., Friedman A., and Genkina A., 2008. *Embedding Trust via Social Context in Virtual Spa-ces*. Available: `http://www.ljean.com/files/NetTrust.pdf`, cit. 2008-09-12.

Chen R., Yeager W., 2008. *A Distributed Trust Model for Peer-to-Peer Networks*, Sun Microsystems. Available: `http://gnunet.org/papers/jxtatrust.pdf`, cit. 2008-09-25.

Fetzer S., 1988. "The World Book Dictionary." World Book Inc., The World Book Encyclopaedia, Chicago, USA.

Gambetta D., 2000. "*Can We Trust Trust?*" In Gambetta, Diego (ed.) Trust: Making and Breaking Cooperative Relations, electronic edition. Department of Sociology, University of Oxford, chapter 13, 213-237.

Lifen L., 2008. "Trust Derivation and Recommendation Management in a Trust Model." In *Proceedings of International Conference on Intelligent Information Hiding and Multimedia Signal Processing* (Harbin, China), 219-222.

Liu Y., Yau S., Peng D., and Yin Y., 2008. "A Flexible Trust Model for Distributed Service Infrastructures." In *Proceedings of the 2008 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing* (Orlando, USA), 108-115.

Mejia M., Pena N., Munoz J., and Esparza O., 2009. "A Review of Trust Modeling in Ad Hoc Networks." *In journal: Internet Research*, vol. 19, Issue 1, 88-104.

Netrvalova A., and Safarik J., 2009. "Interpersonal Trust Model." *In Proceedings of 6$^{th}$ Vienna International Conference on Mathematical Modelling* (Vienna, Austria), pp. 530-537.

Rényi, A., 1961. "On Measures of Entropy and Information." In *Proceedings Fourth Berkeley Symposium Math. Stat. and Probability*, Berkeley, University of California Press, Vol. 1., pp. 547-561.

Ryutov T., Neuman C., and Zhou L., 2007. "Initial Trust Formation in Virtual Organizations." In *International Journal of Internet Technology and Secured Transactions*, vol.1, no. 1-2, 81-94.

Velloso P., Laufer R., Duarte O., and Pujolle G., 2006. HIT: *A Human-inspired Trust Model*. In: IFIP International Federation for Information Processing, Volume 211, ed. G. Pujolle, Mobile and Wireless Communication Network, (Boston Springer), pp. 35-46.

Wang X., Zhang F., 2008. *A New Trust Model Based on Social Characteristic and Reputation Mechanism for the Semantic Web*. International Workshop on Knowledge Discovery and Data Mining. The University of Adelaide, Australia, pp. 414-417.

Wang Y., Varadharajan V., 2007. *Role-based Recommendation and Trust Evaluation*. In: Proceedings 9th IEEE International Conference on E-Commerce Technology and the 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services, Tokyo.

Wu X., He J., and Xu F., 2008. "An Enhanced Trust Model Based on Reputation for P2P Networks." In I*EEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing* (Taichung, Taiwan), 67-73.

Zhang Z., Zhou M., and Wang P., 2008. "An Improved Trust in Agent-mediated e-commerce." *International Journal of Intelligent Systems Technologies and Applications*, vol. 4, 271-284.

## BIOGRAPHY

**ARNOSTKA NETRVALOVA** was born in Plzen, Czech Republic. She is senior lecturer in Department of Computer Science and Engineering at Faculty of Applied Sciences of University of West Bohemia. She holds a M.Sc. in Computer Science from University of West Bohemia in 1977. Her research in modelling and simulation covered simulation of temperature processes in man, and trust modelling. E-mail address is `netrvalo@kiv.zcu.cz`, web page is `http://www.kiv.zcu.cz/~netrvalo`.

**JIRI SAFARIK** was born in Kromeriz, Czech Republic. He received his Ph.D. degree from Slovak University of Technology in 1984. Currently, he is professor in Department of Computer Science and Engineering at Faculty of Applied Sciences of University of West Bohemia. His research covers distributed systems, distributed and parallel simulation. E-mail address is `safarikj@kiv.zcu.cz`, web page available at `http://www.kiv.zcu.cz/~staff`.

# AGENT BASED SIMULATION

# EXPERIMENTAL STUDY OF AGENT POPULATION MODELS WITH A SPECIFIC ATTENTION TO THE DISCRETIZATION BIASES

Pierre Chevaillier, Stéphane Bonneaud, Gireg Desmeulles and Pascal Redou
LISyC: Computer Science Laboratory for Complex Systems
ENIB: Brest National School for Engineers
CERV - 25, rue Claude Chappe, Plouzané, France
email: {chevaillier|bonneaud|desmeulles|redou}@enib.fr

## KEYWORDS

Agent-based simulation, simulators, model design, model evaluation, sensitivity analysis

## ABSTRACT

Solving conceptual models with multi-agent models requires assumptions in terms of discretizations of space, time and in agents. Such implementation choices can introduce biases into simulation results and cause computational models to become non consistent with conceptual models. The goal is to analyze some of those assumptions and the model sensitivity to the corresponding biases through a systematic and experimental study. The models used in this article are specific agent communities, yet parsimonious in parameters, in the perspective of abstracting the results. The study focuses on the extent to which the implemented models agree with the expected outputs, formulated as their precision and accuracy. Results show that simulation tools must enable 1° domain experts to explicit their models (dependencies between behaviors, initializations, discretizations) and 2° systematic analysis of computational biases.

## INTRODUCTION

The translation of conceptual models into agent-based computational models introduces a number of issues to the field of individual-based simulation (Goldspink 2002, Michel et al. 2003). For domain experts, the point is to build and experience their models through simulation (Davidsson 2002) in order to learn from them (Frigg and Hartmann 2006). This led many to the concept of virtual laboratories, which is based on the organization of the modeling activity around simulation (Ramat and Preux 2003). Yet, such concept needs the implemented model to be consistent with the domain expert's conceptual model. Thereafter, the computer scientist's responsibility is not to qualify the representations built by domain experts –thematic and conceptual models– but to operationalize those models (Drogoul et al. 2003).

This operationalization into multi-agent systems brings in assumptions on agents behaviors and implies discretizations of space, time and in agents. If there are no theoretical "good choices" for all models, as described by Michel et al. (2001) for agent scheduling, each assumption is a potential source of biases in the results (Kubera et al. 2009), e.g. spatial and temporal assumptions (Axtell 2000), biases to which the models are more or less sensitive. Various analysis of computational models (Sargent 2004) and especially of individual-based models (IBM) were proposed (Ginot et al. 2006, Santner et al. 2003). Methods for numerical consistency analysis enable to investigate if model implementation affects results in a negligible and controlled manner or not. Yet, it is still necessary to conduct sensitivity analysis to evaluate the impact of implementation choices on models, especially when it comes to predictive models for which there is no reference data for calibration. Computer scientists are responsible for designing tools enabling domain experts to identify the sources of biases introduced in computational models, so that they can quantify their impact on the results (Edmonds 2001, Souissi et al. 2004).

The goal of this article is to analyze the potential sources of biases in agent-based computational models. Based on the *pattern* oriented approach (Grimm et al. 2005), this analysis is pragmatic: it consists in systematically measuring the bias sources using the same theoretical modeling framework, which is both parcimonious in assumptions and parameters and characteristic of a large family of models. Each source of biases is the object of a specific study, for which we exhibit the minimal model and the situations that produce the most demonstrable results. *Patterns* enable all along the article the distinction between results, i.e. the designation of the "good" computational models –what Goldspink (2002) called internally valid models– and the evaluation of introduced biases.

The conceptual framework of this study is theoretical ecology: communities of living species inhabite a homogeneous habitat of finite dimension and limited carrying capacity. Models of agent communities are relevant here, as they can deal with interaction and scale heterogeneities. The study starts with the issue of implementing agents having independent behaviors. Then, we cover the issues of the discretizations brought on by agents, space and time and the model's sensitivity to the corresponding biases. But, before all, we identify the *patterns* and the conceptual model of communities that we want to implement.

## MEANS

The analysis is based on the following principles: the consistency of a computational model with a conceptual model is evaluated as its capacity to account for the *patterns* characterising the conceptual model. Following the *pattern oriented approach*, *patterns* are formulated by domain experts with no supposition on the mechanisms governing the system. Here, we use them as a description of the expected, or theoretical, behavior of the conceptual model. To conduct this study, we focused on *patterns* that could be characterized using quantitative indicators, making no supposition on the general nature of *patterns*.Thereafter the gaps between the computational model's outputs and the *patterns* are the biases on which is focused the analysis. Those biases are analyzed through the *accuracy* and the *precision* of the models' outputs. The *accuracy* is estimated as the gap between the theoretical value and the observed value, and the *precision* as the variability of this observed value. At last, the *sensitivity* is the variation in biases depending on the parameters of the models.

### *Patterns* and indicators

The communities taken in example here are theoretical and the *patterns* that characterize them are therefore somewhat arbitrary. Two *patterns* are considered. *P1: numerical stability*, the population size of each species is constant in time. *P2: spatial homogeneity*, each species inhabits uniformly its habitat. *P1* is characterized by the stability of the variable $N(t)$, the agent population size in time. The effective indicator is $\sigma_N$, standard deviation of $N(t)$ all along the simulation time. Pattern *P2* depends on the population's local density. It is characterized by the homogeneity of this density inside the species habitat. Its calculation is function of the environment grid $M$.

The model *accuracy* is estimated here by the gap between the expected number of agents $N_i$, which is $N(0)$ for a population at equilibrium, and the mean population size $\mu_N$: $\Delta_N = |N_i - \mu_N|$. The standard deviation of $N$, $\sigma_N$, measures the absolute *accuracy* of the model and the variation coefficient $cv_N = \sigma_N/\mu_N$ its relative *accuracy*. Model sensitivity is evaluated by the changes in $\Delta_N$ and $cv_N$ according to the implementation choices.

### Conceptual model

The processes structuring the system dynamic are basic: agents breed, die, move around their habitat and hunt other agents. Their habitat is simple: it's a homogeneous disk (with the exception of its edges) positioned at the center of the environment. The edges are smoothed and describe an affinity gradient going from the maximum (preferential habitat) to the minimum (a place not fit to live in). The habitat is perceived by agents through the distribution of the affinity in the environment: $\alpha_H(x, y) \in [0, 1]$ is the affinity to habitat $H$ at location $(x, y)$.

## AGENT BEHAVIORS

The study focuses here on the implementation in a single agent of conceptually independent behaviors. The question is to know if those behaviors can be independent in the computational model. The implementation lies on choices on the deterministic or stochastic nature of behaviors, on their synchronization, on state variables management and communities initialization. To carry out this study, let's consider the case of a population "at equilibrium", in which as many individuals die in a given period as others are born. This corresponds to *pattern* $P1$; the structuring processes are the mortality and reproduction.

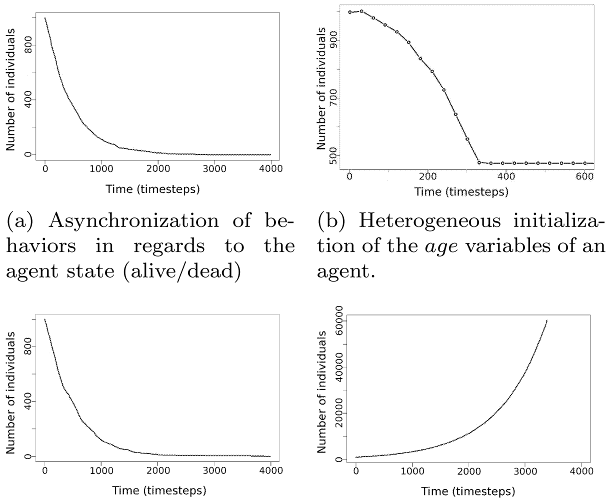
### Deterministic model

The simplest deterministic computational model for the reproduction is if an individual creates a new born when it reaches its species age of sexual maturity; symmetrically, an individual dies when its age reaches its species age of longevity. A specific solution, consistent with the *pattern*, is to set the age of sexual maturity to the longevity. Each agent has two behaviors, function of its age, activated at each timestep. Given such simple implementation, let's now consider the implementation choices that need to be taken in order for this solution to reproduce *pattern* P1.

### Behavior synchronization

Behaviors are said to be *independent* if they can be executed in any order, without the execution order affecting agent states, which is equivalent to a synchronous activation. Two properties must be verified: 1° the execution of one does not impact the execution of others, 2° the system dynamic is independent of the execution order. Let's analyze those properties with the reproduction and mortality behaviors.

Consider the solution where behaviors modify disjoined sets of state variables. This is equivalent, in the present case, with having a variable age per behavior, both variables evolving in an identical manner in each behavior. Executing the reproduction does not influence the mortality's execution, whereas mortality makes reproduction impossible. The result, not consistent with *pattern* $P1$, is an exponential decrease of the population size (figure 1a), because, here, the execution of one of the



(a) Asynchronization of behaviors in regards to the agent state (alive/dead)

(b) Heterogeneous initialization of the *age* variables of an agent.

(c) Modification of age at the beginning of the mortality behavior

(d) Modification of age at the end of the mortality behavior

Figure 1: Sensitivity to behaviors synchronization, to related variables and to initial states.

behaviors impacts the execution of the other and therefore the system dynamic is dependent of the execution order. If a mechanism ensuring the equity of behavior executions is added, that is a mechanism that guarantees that all behaviors of an agent are executed as many times as one another, the model accurately reproduces *pattern P*1.

Let's now consider the evolution of the age and the case of mortality. This behavior executes two elementary actions one after another: increment the age and change the agent state for "dead". Depending on the order those actions are executed by the behavior and again if no mechanism ensuring the equity of behavior executions is added, then the results are those of figures 1c or 1d. Here again, the global dynamic is all the way different.

Identifying the issue and its solution is here obvious, but it is less the case in general, when behaviors do not execute themselves at the same frequency, when behaviors side effects are indirect or when the implementation is less simple.

### Initialization of state variables

The initialization of IBMs is a critical issue for their simulation (Goldspink 2002) and the initial state of each agent cannot be directly set using observed macrosocpic data. The solution is for domain experts to define random initializations of state variables according to probability laws, parametric or not. Yet, what about the initialization of the various variables of a same agent?

Take again the model of the reproduction and mortality processes, with each their own variable age and a mechanism ensuring that their execution is fair (equity of execution). Here, the two variables are joined—conceptually corresponding to the same idea, for demonstration sake—and need to be initialized. It is obvious here that the two variables must have the same initial value and if this is not guaranteed, we might get the dynamic in figure 1b: disappearance of about half of the population. Such dynamic was obtained by randomly (using an uniform distribution), and independently, initializing variables of all agents. Thereafter, about half of the agents had an initial value for the mortality's variable that was greater then the one for the reproduction's variable.

Here again the solution for this specific implementation is obvious, yet the general solution for this issue would require to know the matrix of variances-covariances for all state variables of agents, which is, by essence, impossible in an IBM as state variables are model artefacts, inaccessible—directly, or even through proxys—in the natural system. Processing statistically those covariances is impossible for a model with many parameters and few observed data. Establishing analytically them would require to have an analytical solution to the problem, which is unattainable with a complex system.

### Stochastic behaviors

A behavior is stochastic if it executes itself with a specific probability or if its outputs are stochastic. Using a stochastic agent-based computational model is equivalent to reproducing within each individual the non deterministic nature of the system. In addition to the previously identified restrictions, behavior independence requires independent execution probabilities, assuming that random number generators are independent too and that the execution of one does not impact the others.

Consider here behaviors which results are non continuous: the creation or destruction of an agent, but this would be the case with any other non continuous change in the agent state. Such behavior makes the system dynamic sensitive to the number of agents, as shown in table 1. Here, the accuracy of the result is very poor below 100 agents. Figure 2 shows that the decrease of the variation coefficient is linear against the number of agents on a log-log scale. In this example, the absolute benefit of accuracy is low beyond 500 agents. However, figure 3 shows that the 30 trajectories out of which was processed the mean dynamic for 1000 agents are quite different, even though the global bias is low.

### Analysis

Those examples show the strong sensitivity of the system to implementation artefacts with very strong biases for some of them. If general solutions exist for synchronizing independent behaviors, it is not the case for managing and initializing variables. Furthermore, stochastic

Table 1: Stochastic model: sensitivity to the number of agents. $\hat{\mu}_N$ and $\hat{\sigma}_N$ are calculated on the mean dynamic of 30 simulations for a given $N_i$.

| $N_i$ | $\hat{\mu}_N$ | $\hat{\sigma}_N$ | $cv_N$ |
|---|---|---|---|
| 10 | 12 | 42.2 | 3.47 |
| 50 | 30 | 53.6 | 1.78 |
| 100 | 98 | 103. | 1.05 |
| 500 | 499 | 296. | 0.59 |
| 1000 | 1005 | 327. | 0.33 |
| 5000 | 4981 | 777. | 0.16 |

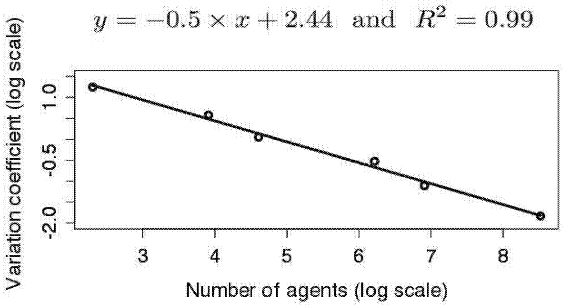


Figure 2: Stochastic model: evolution of $cv$. $y = \ln(cv_N)$ ; $x = \ln(N_i)$

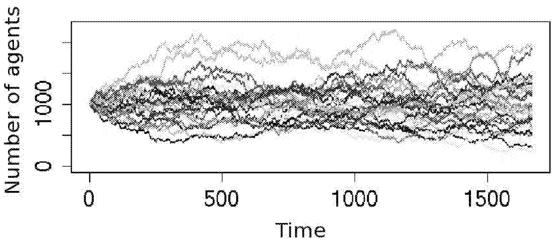


Figure 3: Stochastic model: trajectories of 30 simulations (1000 agents).

Table 2: Discretization in agents, case of one population and deterministic density-despendent behaviors: effect of density, $K_{max}$.

| $K_{max}$ | $N_i$ | $\mu_N$ | $\sigma_N$ | $cv$ |
|---|---|---|---|---|
| 0.64 | 50 | - | - | - |
| 1 | 77 | 87 | 4.89 | $6.10^{-2}$ |
| 6 | 467 | 537 | 10.7 | $2.10^{-2}$ |
| 13 | 1013 | 1110 | 11.2 | $1.10^{-2}$ |
| 65 | 5067 | 5211 | 18.2 | $4.10^{-3}$ |
| 129 | 10057 | 10227 | 19.8 | $2.10^{-3}$ |

models require high numbers of agents and the execution of many simulations. This increases processing time and makes the analysis of results complicated (Bertelle et al. 2007). It is therefore essential to characterize *patterns* with very synthetic indicators that have minimal sensitivity.

## DISCRETIZATION IN AGENTS

The study now addresses the discretization in agents of the model. The choice of such granularity is taken according to the number of agents required to describe a given phenomenon. Choosing the “right” amount of agents to describe a given population is a compromise between available resources and the minimum accuracy and precision required.

### Homogeneous discretization: the case of one population

Agent modeling introduces a non continuous dynamic in number of the population, like other IBM approaches. However, the number of agents in the simulation does not always correspond to the amount of individuals of studied systems that can be composed of a huge amount of individuals. Thereafter, using agent populations to represent a much larger population of individuals can introduce a bias.

**Computational model.** Consider again the case of a population “at equilibrium”, which is now spatialized.

*Reproduction behavior.* Deterministic and based on the ecological notion of sexual maturity, it is implemented using an accumulator $\Phi$. Such accumulator depends on a coefficient $\tau_\Phi$: $\Phi$ is incremented at each execution by $\alpha_H(x,y) \times \tau_\Phi$. The number of children per agent to create is then, at each execution, the floor value of $\Phi$.

*Mortality behavior.* Deterministic, it is based on the affinity $\alpha_H(x,y)$ and $K_{max}$, which is a proxy parameter for the carrying capacity of the habitat. The agent dies if the local density in agents is greater than $K_{max}$ weighted by the local affinity to the environment.

*Recruitment behavior.* The location $(x,y)$ of a newly created agent (resulting from the reproduction behavior) is calculated using a non-parametric probability distribution based on the affinity to the environment $\alpha_H(x,y)$. The positionning of the new agent at this location is the result of the execution of the recruitment behavior, which is executed once when the agent is created.

**Mean.** Discretization $M$ equals $16 \times 16$ cells. The $\Phi$ variables are initialized to zero in all agents. Finally, the varying factor is the parameter of maximum density $K_{max}$ that controls the number of agents the habitat can host.

**Results.** Results from table 2 indicate that the model reproduces *pattern* $P1$, except of course for the first simulation ($K_{max} < 1$). Figure 4 shows that the increase of the number of agents implies a decrease of $cv$ and that this decrease is linear on a log-log scale. At last, the model reproduces more or less *pattern* $P2$ as shown in figure 5: a too small number of agents produces a spatial distribution distant from the expected theoretical distribution.

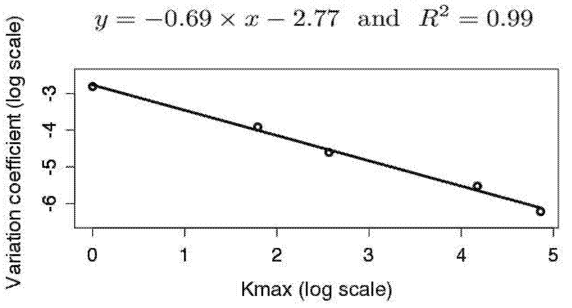
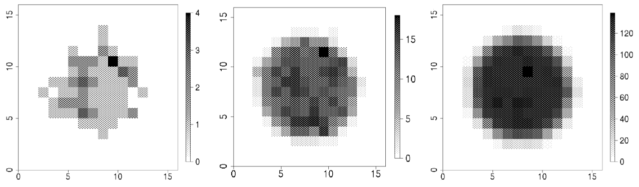


Figure 4: Discretization in agents, case of one population: evolution of the variation coefficient $cv$ according to $K_{max}$. $y = \ln(cv_N)$ ; $x = \ln(K_{max})$



(a) $K_{max} = 1$ (b) $K_{max} = 13$ (c) $K_{max} = 129$

Figure 5: Discretization in agents, case of one population: spatial distributions of the population according to $K_{max}$ (t=2000).

### Heterogeneous discretization: the case of several populations

If the choice of a granularity must be "adapted" to each phenomenon to represent, it can become necessary to use different granularities for different populations in the same model. The study addresses here this heterogeneity of scale that introduces issues regarding the interactions between components of the model described differently in terms of number of agents. To conduct this study, let's consider two populations of preys and predators "at equilibrium" and in trophic interaction.

**Computational model.** To enable the interaction of two populations with different discretizations in agents, it is necessary to introduce a scale factor $\rho$. This factor describes the granularity in number of agents of a population. It is the ratio between the number of individuals $X$ in the real population and the number of agents in the simumation $N$: $\rho = X/N$. Let's add $\rho$ to the model of the previous study and an additional predation behavior describing the interactions between the different types of agents (preys and predators). All agents are therefore associated to reproduction, mortality and recruitment behaviors. Only predators execute a predation behavior.

This last behavior depends on two parameters, $Q_{max}$ and $Q_{min}$, that are respectively the amount of preys that a predator agent tries to eat (the predation impact on preys) and the minimum amount that this agent needs to eat to survive (the predation impact on predators). $Q_{max}$ and $Q_{min}$ are expressed in number of individuals.

Table 3: Discretization in agents, case of two populations: results with $X_p$ the number of prey individuals, $X_P$ the number of predator individuals and $\Delta_X = |X_i - \mu_X|$.

| $N_{p,i} - \rho_p$ | $\Delta_{X_p}$ | $\sigma_{X_p}$ | $N_{P,i} - \rho_P$ | $\Delta_{X_P}$ | $\sigma_{X_P}$ |
|---|---|---|---|---|---|
| 5000 – 200 | 21657 | 3271 | 5000 – 2 | 3 | 13 |
| 1000 – 1000 | 163683 | 24590 | 1000 – 10 | 438 | 55 |
| 500 – 2000 | 222350 | 32292 | 500 – 20 | 516 | 104 |
| 100 – 10000 | 368750 | 46010 | 100 – 100 | 2871 | 399 |
| 5000 – 200 | 10985 | 90353 | 1000 – 10 | 10000 | 0 |
| 5000 – 200 | 9030 | 70540 | 500 – 20 | 10000 | 0 |
| 5000 – 200 | 175928 | 50308 | 100 – 100 | 7880 | 500 |
| 1000 – 1000 | 42575 | 5506 | 5000 – 2 | 5 | 11 |
| 500 – 2000 | 45717 | 9935 | 5000 – 2 | 7 | 13 |
| 100 – 10000 | 182417 | 33632 | 5000 – 2 | 4 | 12 |

At last, $\rho_p$ and $\rho_P$ are respectively the preys' scale factor and the predators' scale factor. Thereby, a predator tries to eat a maximum of $Q_{a,max} = Q_{max} \times \rho_P/\rho_p$ agents and a minimum of $Q_{a,min} = Q_{min} \times \rho_P/\rho_p$ agents. The predator hunts in its cell, where it randomly captures a prey, if it encounters one, until it has eaten $Q_{a,max}$ agents or if the cell is empty. Last, the predator dies if it has not eaten $Q_{a,min}$ agents and if it does not find more preys.

**Mean.** Discretization $M$ equals 16 × 16. The scale factor being explicit, the number of prey individuals is $X_{p,i} = 10^6$ and the number of predator individuals is $X_{P,i} = 10^4$. The varying factors are the number of agents and the corresponding scale factors: $N_p$, $N_P$, $\rho_p$ and $\rho_P$.

**Results.** The first four lines of table 3 show that the bigger the scale factor, the bigger the gap between the number of individuals the agents were to model and the results: the accuracy is very bad and we can have $\mu_{N_p} \ll 10^6$ and $\mu_{N_P} \ll 10^4$. The lost of precision depends on the number of agents and the heterogeneity of scale induces biases that are generally stronger that in the homogeneous case. Especially, the three simulations for which $\rho_p$ is minimal ($\rho_p = 200$) and $\rho_P$ is greater than 2 give dramatic $\Delta_{X_P}$. Last, simulations for which $\rho_P$ is minimal ($\rho_P = 2$) and $\rho_p$ is greater than 100 give good results. In conclusion, simulations with 5000 predator agents give very good results whatever the number of preys.

### Analysis

Globally, increasing the number of agents improves the accuracy and precision of the results, which was expected. However, the study of the homogeneous discretization shows that this increase is not linear and, asymptotically, the benefit is null. The results on *pattern P*2 take the same direction. Nonetheless, a too small number of agents per cell leads to results that do not correspond to the conceptual model, but this is a consequence of a non adapted discretization in agents.

The study of the heterogeneous discretization shows that the model is very sensitive to this factor and to its heterogeneity: biases can be very high. A small discretization in number of preys seems to less influence the results.

## SPATIAL DISCRETIZATION

The study focuses now on the discretization of the environment needed in the computational model. Indeed, information about the environment is discretized and their perception by agents is thus biased in: 1° the relation of the agent with its environment and 2° the relation of the agent with others.

### Homogeneous discretization: interaction with the environment

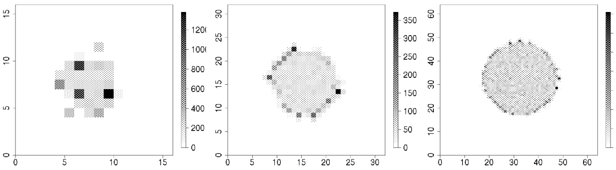
The question here is to know how the environment discretization influences the spatial dynamic of an agent population. To achieve this study, let's consider a population "at equilibrium", roaming in its habitat and distributing itself homogeneously. This situation corresponds to *pattern* $P2$: the structuring process is migration. Processes of reproduction, natural mortality and predation are ignored in this particular study.

**Computational model.** The migration behavior is the tendency for an agent to follow the local affinity gradient in order to reach an "ideal place", which is a trade-off between the highest affinity and a low density. The agent calculates a migration vector that is the sum of: 1° the vector towards the neighboring cell with the highest affinity and 2° the vector opposite the neighboring cell with the highest density. The agent moves in the resulting direction of a given migration step that is smaller than the size of the smallest grid cell. Thereby, the roaming does not depend on the discretization except through the perception of the environment. In these simulations, the perception is based on a Von Neumann neighborhood.

**Mean.** We have $N_i = 10^4$ and the varying factor is the discretization of the information the agent perceive in the environment: the affinity and the density.

**Results.** Figure 6 shows the various resulting distributions at timestep 3600 and shows that *pattern* $P2$ is more or less reproduced. Recall that a strong discretization induces a small number of agents per cell, which partly explains the differences in the results, however, the model sensitivity to the studied factor is high.

### Homogeneous discretization: interaction between agents



(a) $M = 16 \times 16$ (b) $M = 32 \times 32$ (c) $M = 64 \times 64$

Figure 6: Spatial discretization, interaction with the environment: population distributions according to $M$.

Here we focus on the influence of the spatial discretization on the relation between agents. Let's consider the model of the previous study to which is added a population of predators that hunt the first population wich is now the preys. Again, *pattern* $P2$ is considered and the structuring processes are migration processes.

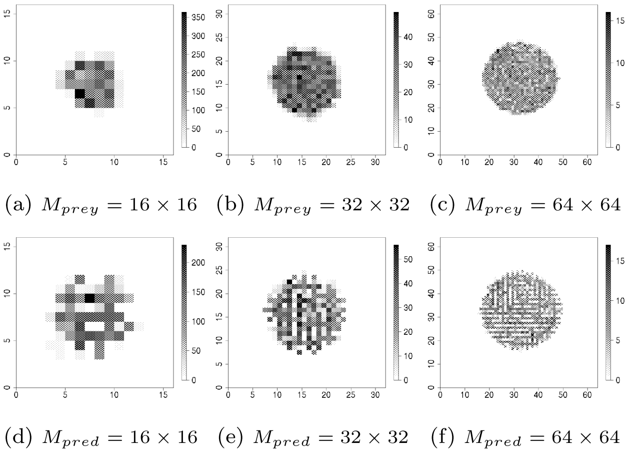
**Computational model.** The migration behavior described in the previous study is now the migration of preys. It is slightly modified to take into account predators: to compute its migration vector, a prey considers a third vector that is opposite to the neighboring cell of highest density in predators. Thereby, preys tend to flee predators. Concerning the migration behavior of predators, the "ideal" habitat of a predator is locally characterized by the highest density in preys and the lowest in predators. A predator agent computes a migration vector that is the sum of the vector to the neighboring cell of lowest intraspecific density and the vector to the neighboring cell of highest density in preys. A predator is also drawn by areas of highest affinity, but only when it has no preys in sight, which is a borderline case.

**Mean.** We have $N_{prey,i} = N_{pred,i} = 5.10^3$ and again the varying factor is the spatial discretization.

**Results.** Figure 7 shows the resulting distributions at timestep 3600 and shows, as for the case of one population, that *pattern* $P2$ is more or less reproduced. For predators, the homogeneity is less pronounced and agents seem to get organized in strips. Yet, such strips do not depend on the discretization, they are constant through out the different results and are therefore not relevant to our study. What is important are the inconstant elements in the results that thus depend on the discretization. The greater the discretization (smaller grid cell sizes), the more homogeneous seems the distribution. But, this is to put in perspective again with the fact that a high discretization leads to a low number of agents per cell. We simply notice that a strong discretization produces an agent distribution that seems consistent with the conceptual model.

### Heterogeneous discretization

Following the logic of the previous study: depending on the needs in precision of studied populations, it can become necessary to have different spatial discretizations for the different populations. The question here is therefore to study what kind of impact and multi-scale interaction are introduced by such heterogeneity of description. To achieve this study, let's consider again the computational model of the previous study.



(a) $M_{prey} = 16 \times 16$ (b) $M_{prey} = 32 \times 32$ (c) $M_{prey} = 64 \times 64$

(d) $M_{pred} = 16 \times 16$ (e) $M_{pred} = 32 \times 32$ (f) $M_{pred} = 64 \times 64$

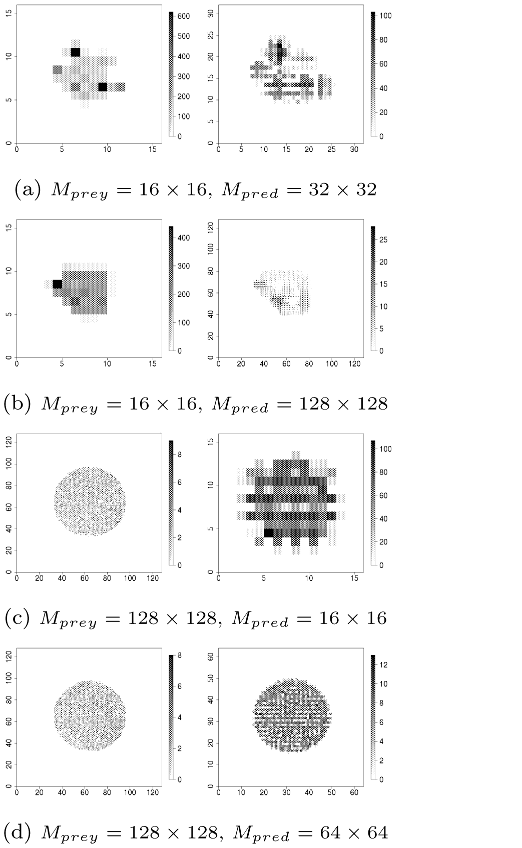
Figure 7: Spatial discretization, interaction between agents: preys distributions (on top) and predators (on the bottom) according to $M$.

**Mean.** Consider two populations, each in a different environment. To do this, we use the *virtuOcean* simulation platform that enables the building of two models using *model-agents* to couple them (Bonneaud et al. 2007). Last, we have $N_{prey,i} = N_{pred,i} = 5.10^3$.

**Results.** Figure 8 shows the resulting distributions at timestep 3600. The results are much less convincing than in the homogeneous case: they are even more influenced by the spatial discretization. Especially, a low discretization of the preys environment leads to non homogeneous distributions that do not depend on the habitat. The preys and predators distributions of figure 8b are for instance very different from the habitat theoretical distribution. On the opposite, a strong discretization of the preys environment leads to very good results in regards to the theoretical distribution and the results in the homogeneous case, whatever the spatial discretization of the predators environment.

## Analysis

Homogeneous spatial discretization impacts the population distribution differently depending on the discretization used. The model being minimal, it explains in itself some density peaks that are then (temporary) stable locations for the agents. But the model globally behaves very differently depending on the discretization. Concerning the preys-predators case study, here again the results tend to show that the system is sensitive to the spatial discretization. Distributions are however more homogeneous and regular and globally the system behaves consistently with the conceptual model. This shows that interaction processes in between populations can hide biases impacts due to the spatial discretization. Last, the study of heterogeneous spatial discretization shows a strong sensitivity of the system to this factor and the resulting dynamics are further away from the expected dynamics than in the homogeneous case. In particular, a low discretization results in population distributions that are non consistent with the conceptual model, while a strong discretization leads to distributions that are consistent.



(a) $M_{prey} = 16 \times 16$, $M_{pred} = 32 \times 32$

(b) $M_{prey} = 16 \times 16$, $M_{pred} = 128 \times 128$

(c) $M_{prey} = 128 \times 128$, $M_{pred} = 16 \times 16$

(d) $M_{prey} = 128 \times 128$, $M_{pred} = 64 \times 64$

Figure 8: Spatial heterogeneous, interaction between agents: spatial distributions of preys (on the left) and predators (on the right) according to $M$.

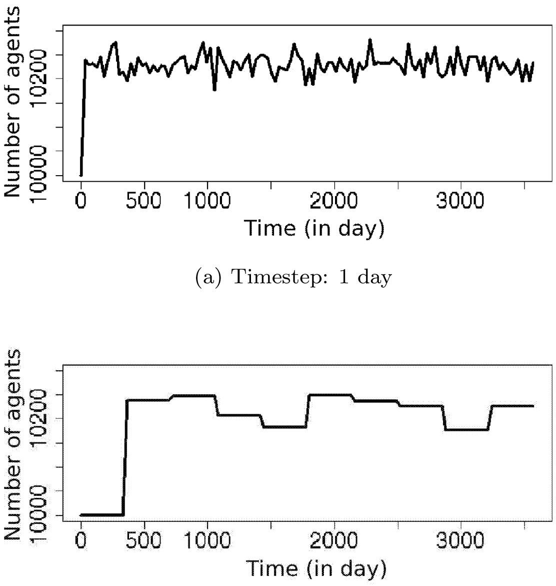
## TEMPORAL DISCRETIZATION

The study now addresses the impact of the temporal discretization on the results. The question is even more relevant for agent-based models and more generally IBM: a high activation frequency of agents seems to help refine the dynamic of their behaviors. But how does that work? What type of impact on the results does such a choice in the discretization induces?

**Computational model.** To achieve this study, we focus again on a population "at equilibrium" and especially on agent creation and destruction processes, that produce non continuous changes in the population's state and a model sensitive to the timestep. Let's consider again the computational model used for the study of the homogeneous discretization in agents (case of one population) with the reproduction, mortality and recruitment behaviors. We now need to take into account the temporal scale: only the reproduction behavior de-

Table 4: Temporal discretization: results.

| $dt$ | $\tau_\Phi$ | $\mu_N$ | $\sigma_N$ | $cv$ |
|---|---|---|---|---|
| day | $3.10^{-3}$ | 10229 | 20.6 | $2.10^{-3}$ |
| month | 0.09 | 10226 | 30.0 | $3.10^{-3}$ |
| semester | 0.54 | 10216 | 52.2 | $5.10^{-3}$ |
| year | 1.08 | 10198 | 70.7 | $7.10^{-3}$ |



(a) Timestep: 1 day

(b) Timestep: 360 days (1 year)

Figure 9: Temporal discretization: evolution of the number of agents according to the timestep.

pends on the timestep and is to be modified. Indeed, $\tau_\Phi$ implicitly depends on the time and therefore has a unit. For a change in the timestep to influence the system, one must be able to convert $\tau_\Phi$ depending on the timestep. Thereafter, $\tau_\Phi$ is expressed by: $\tau_\Phi^u = \tau_\Phi^{day} \times c_{day,u}$, with $c_{u_1,u_2}$ the scale factor or conversion unit from $u_1$ to $u_2$, the unit of reference being the day.

**Mean.** Let's consider the evolution of a population of agents on a period of ten years (3600 days). $M$ equals $16 \times 16$, $N_i = 10^4$ and the variables $\Phi$ are set to 0 in the initial population. The starting assumption is: $\tau_\Phi^{day} = 3.10^{-3}$. The varying factor is the timestep, to which is associated a corresponding $\tau_\Phi$.

**Results.** Table 4 shows the studied timesteps, the corresponding $\tau_\Phi$ and the results. First, the model reproduces *pattern* $P1$. This is corroborated by figure 9 which shows two examples of evolution of the number of agents: the dynamics are very similar (except for the variation frequency, which is obvious). At last, figure 10 shows the evolution of $cv$ depending on the timestep. We notice a linear decrease of $cv$ when the timestep decreases.

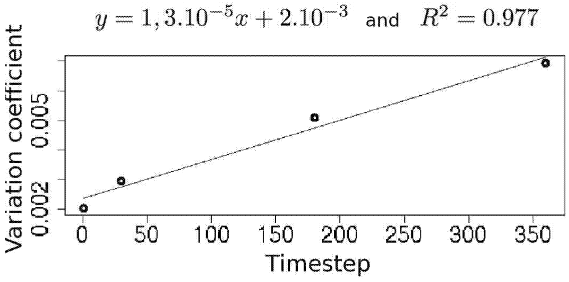


Figure 10: Temporal discretization: evolution of $cv$ according to the timestep. $y = cv_N$ ; $x = dt$

## Analysis

The study of the temporal discretization shows that the benefit in precision is here linear. Furthermore, and it is an expected result, between a timestep of a year and a timestep of a day, the precision is not dramatically improved. Lastly, the system's sensitivity to the temporal discretization might be more critical with other models, with populations of roaming agents for instance. A low activation frequency of behaviors would induce great migration amplitudes.

## CONCLUSION

Dependence between behaviors, their synchronization and the choice of initial states are critical implementation issues. Dependent variables can for instance break the model's consistency apart. Moreover, the stochastic solution as an operational tool leads to implicit assumptions that must be clearly known.

The sensitivity analyses of discretizations presented in this article often exhibit predictable results, yet some are more critical than excepted. Globally, the models are very sensitive to those factors and, depending on the discretization, the results can be non consistent with the conceptual model. If some results are intuitive and the sources of biases known, it is impossible to know in general their consequences on the results. Depending on the models, the biases are more or less easily noticeable and some processes can even partly hide them. When considering complex models, such biases can lead to wrong interpretations of the results. In consequence, their study should be systematic.

Building a model and choosing its granularity means *de facto* making a compromise between the required precisions of the results and available resources. This compromise depends on the system, the resources and the implemented model. Furthermore, the implementation choices are justified in regards to an application and its requirements in terms of precision and accuracy. There are no absolute acceptable thresholds for a model's accuracy and precision, yet the latters should be known in order to be able to understand the results and interpret them. It is therefore necessary to have specialized simulation tools that enable systematic analysis of computational biases. The perspective of this work is to propose a modeling language for domain experts to explicit their models (behaviors, variables, discretizations), with an operational semantic that guarantees the properties we have identified here.

**Thanks**

This work is part of the Chaloupe project financed by the French National Agency for Research (`http://www.projet-chaloupe.fr`) and has been financially supported by the Région Bretagne.

## REFERENCES

Axtell R., 2000. *Effects of Interaction Topology and Activation Regime in Several Multi-Agent Systems.* In *MABS*. 33–48.

Bertelle C.; Dutot A.; Guinand F.; and Olivier D., 2007. *Organization Detection for Dynamic Load Balancing in Individual-based Simulations. Multiagent and Grid Systems: special issue on Nature-Inspired Systems for Parallel, Asynchronous and Decentralized Environments*, 3, no. 1, 141–163. ISSN 1574-1702 (Print) 1875-9076 (Online).

Bonneaud S.; Redou P.; and Chevaillier P., 2007. *Oriented pattern agent-based multi-modeling of exploited ecosystems.* In *EuroSim07*.

Davidsson P., 2002. *Agent Based Social Simulation: A Computer Science View. Journal of Artificial Societies and Social Simulation*, 5, no. 1.

Drogoul A.; Vanbergue D.; and Meurisse T., 2003. *Multi-agent Based Simulation: Where Are the Agents ?* In *MABS*. 43–49.

Edmonds B., 2001. *The Use of Models – making MABS more informative.* In *Proceedings of the second international workshop on Multi-agent based simulation*. 15–32.

Frigg R. and Hartmann S., 2006. *Models in Science. Metaphysics Research Lab.*

Ginot V.; Gaba S.; Beaudouin R.; Aries F.; and Monod H., 2006. *Combined use of local and ANOVA-based global sensitivity analyses for the investigation of a stochastic dynamic model: Application to the case study of an individual-based model of a fish population. Ecological Modelling*, 193, 479–491.

Goldspink C., 2002. *Methodological Implications Of Complex Systems Approaches to Sociality: Simulation as a foundation for knowledge. Journal of Artificial Societies and Social Simulation*, 5, no. 1.

Grimm V.; Revilla E.; Berger U.; Jeltsch F.; Mooij W.; Railsback S.; Thulke H.H.; Weiner J.; Wiegand T.; and DeAngelis D., 2005. *Pattern-oriented modeling of agent-based complex systems: lessons from ecology. Science*, 310, no. 5750, 987–991.

Kubera Y.; Mathieu P.; and Picault S., 2009. *How to avoid Biases in reactive simulations.* In *Proceedings of the 7th International Conference on Practical Applications of Agents and Multi-Agent Systems (PAAMS'09)*. 100–109.

Michel F.; Ferber J.; and Gutknecht O., 2001. *Generic Simulation Tools Based on MAS Organization.* In *Proceedings of the 10 th European Workshop on Modelling Autonomous Agents in a Multi Agent World MAMAAW'2001*.

Michel F.; Gouach A.; and Ferber J., 2003. *Weak Interaction and Strong Interaction in Agent Based Simulations.* In *Lecture Notes in Artificial Intelligence 2927*. 43–56.

Ramat E. and Preux P., 2003. *Virtual Laboratory Environment (VLE): A Software Environment oriented Agent and Object for Modeling and Simulation of Complex Systems. Simulation Practice and Theory*, 11, 45–55.

Santner T.; Williams B.; and Notz W., 2003. *The Design and Analysis of Computer Experiments.* Springer.

Sargent R.G., 2004. *Validation and verification of simulation models.* In *Proceedings of the 2004 Winter Simulation Conference*. 121–130.

Souissi S.; Ginot V.; Seuront L.; and Uye S., 2004. *Handbook of scaling methods in aquatic ecosystems: measurement, analysis, simulations*, CRC Press, Boca Raton, chap. Using Multiagent Systems to Develop Individual-Based Models for Copepods: Consequences of Individual Behavior and Spatial Heterogeneity on the Emerging Properties at the Population Scale. 523–542.

# ACTOR AND OBSERVER PROCESSES OF NORMATIVE AGENTS IN SOCIAL SIMULATIONS

Ulf Lotzmann
Michael Möhring
Institute of Information Systems Research
University of Koblenz
Universitätsstraße 1, Koblenz 56070, Germany
E-mail: {ulf, moeh}@uni-koblenz.de

**KEYWORDS**
Agent-based simulation, norm innovation, social science, emergence, immergence.

## ABSTRACT

This paper describes the design of an agent model for simulating normative behaviour and norm formation processes. The model is based on a scientific theory of norm innovation, provided by the FP6 project EMIL (EMergence In the Loop). The main focus of the paper is the conversion of the theoretical framework towards a software implementation (specified formally by UML-based activity diagrams) that can be applied for multiple simulation scenarios.

## INTRODUCTION

This paper focuses on the design and implementation of an agent-based simulation approach which describes the process of norm innovation. It is part of the FP6 project EMIL (EMergence In the Loop: Simulating the two-way dynamics of norm innovation) as an example of simulating emergent properties in complex social systems, funded by the EU initiative "Simulating Emergent properties in Complex Systems" (no. 033841). The model design is based on a scientific theory of norm innovation, provided by EMIL (Andrighetto et al. 2007). Using this theoretical framework, an agent-based norm innovation model is designed, implemented and tested by using different simulation scenarios.

## THEORETICAL FRAMEWORK

The EMIL project especially focuses on understanding and analysing norm innovation processes in social systems, which can be seen here as a special case of complex systems, composed of many different interacting intelligent autonomous agents. In general, including norms in multiagent models seems to be a promising concept to understand human (and artificial) agent cooperation and co-ordination. Therefore, the design of agents with normative behaviour (i.e. normative agents) is of increasing interest in the multiagent systems research (Boella et al. 2007).

Because of the fact that norms can be seen as a societal regulation of individual behaviour without sheer pressure, special attention in modelling and analysing norm innovation processes should be given not only to the inter-agent behaviour but also to the internal (mental) states and processes of the modelled agents (intra-agent) (Neumann 2008). Following this, the dynamics of norm innovation in EMIL can be described mainly by two processes:

- **immergence**: intra-agent process by means of which a normative belief is formed into the agents' minds (Andrighetto et al. 2008).

If this happens often enough, the resulting behaviour becomes a "sociological phenomenon" (Durkheim 1895):

- **emergence**: inter-agent process by means of which a norm not deliberately issued spreads through a society.

From a more practical point of view, agent architectures, which are principally able to represent mental states and cognitive processes of agents, usually follow the classical BDI (Belief-Desire-Intention) approach (Bratman 1987). Even if an approach like the BOID (Belief-Obligation-Intension-Desire) architecture (Broersen et al. 2001) enhances this concept towards a more norm-like modelling by introducing an additional feature "obligation", the complexity of modelling and simulating norm innovation processes proposed in EMIL does not recommend the usage of such general and predefined architectures. This is because of the fact that all agent knowledge (beliefs, desires, intentions and especially obligations) are seen as predefined and "hard-wired" agent properties. With these structures, it is impossible to answer questions regarding the origin of new norms or the communication of norms between agents (Campenni et al. 2008).

## ARCHITECTURE OF A NORMATIVE AGENT

### Concepts

Based on the theoretical background of agent-based approaches on one hand and principles of norm innovation dynamics on the other: How can an agent be equipped with capabilities that allows norm-oriented behaviour and establishing, perceiving and, extending norms?

Below a possible answer to this question is presented – a framework realizing the theoretical concepts proposed in EMIL. It is also associated with the EMIL project, thus the resulting software has been given the name EMIL-S(imulator).

As usual in agent-based simulation, the communication between agents is based on a concept of messages, which trigger the processing of events agents perceive within the environment in which they are situated and which they influence by corresponding actions. For example, in a traffic simulation scenario a car driver perceives a pedestrian on one side of the road and reacts by slowing down the car. This event motivates the pedestrian to begin with crossing the road, which, in turn, makes the car driver stop the car. These events, originating from an agent's perception, are called **environmental events** in EMIL-S.

The modelling of agent behaviour based on societal regulations (or norms) and the generation of this kind of regulations (e.g. by learning processes) obviously requires some fundamental extensions of the approach described so far:

First of all, the introduction of an additional category of events is necessary, which allows the **assessment of (environmental) events** and corresponding actions, performed in a concrete application scenario. For example, it should be possible to evaluate the behaviour of another agent (e.g. admonish a car driver for jumping a red light) by positive/negative valuations or sanctions, or even advising an agent what to do under a particular condition by "deontic" assertions (e.g. "Do not cross the street until the traffic light switches to green!"). These events are called **norm-invocation events**.

Secondly, these norm invocations should not only be activated by agents directly involved in a concrete situation, but also by agents who observe this situation (e.g. watching an accident). Therefore, EMIL-S distinguishes between two agent roles: **actor** and **observer**.

Using the described environmental and norm-invocation events by actors and observers, the **learning capabilities** in EMIL-S (to form preliminary norms into the agents' minds) can be described as follows:

- **Reinforcement**: learning from an agent's own experience (e.g. a pedestrian has a near-collision with a car because of not using the striped area for crossing a street)
- **Imitation**: learning by observing other agents' experience (e.g. observing a near-collision between a pedestrian and a car because this pedestrian did not use the striped area for crossing a street)
- **Normative learning**: listening to other agents' reports of their experiences (e.g. "You should use the striped area for crossing a street!")

### Basic Structures

*Agents.*

The abovementioned processes require at least two agent-internal memories:

- **Event boards** memorizing the history of incoming environmental events including the conducted actions. For this purpose each agent has an event board for his/her own perceptions as well as additional event boards for each observed agent.
- **Normative frame** holding preliminary norms, derived from the experiences logged in the event board during the simulation.

Furthermore, each agent needs an initial set of rules (**Initial Rule Base, IRB**), which contains rules describing basic behavioural elements, and thus constituting the seeds for more complex rules emerging from the simulation process.

The event board is a chronologically sorted sequence whose elements contain the following data:

- the environmental message;
- the current (environmental) agent state (e.g. velocity and perception range for a car driver);
- the associated action tree (see Figure 1) with the individual selection probability function.

For each subsequence of the event board a so-called Classifier (CLA) can be generated. It allows comparisons between event board subsequences and normative frame entries in the norm formation process later on.

Each event board sequence describes a consecutive fragment of agent behaviour, thus introducing a higher level of complexity. It must be assumed that only within this complexity level, regularities and in particular norms are residing.

An entry of the normative frame, which can be given in advance by the modeller, or which arises from the detection of regularities in event board sequences, contains the following elements:

- the associated classifier of the event board sequence;
- the events from the corresponding event board sequence;
- a generated rule (a merged action tree);
- a valuation history, holding a statistical report of the valuations received on the respective rule.

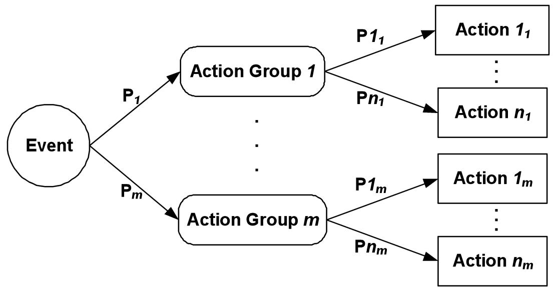


Figure 1: Event Action Tree

Finally, each agent must be equipped with a set of initial rules (IRB), which allows it to act in the simulation environment. Rules in EMIL-S are represented as so-called **event-action trees** (Figure 1), a kind of decision trees that represent the dependencies between events and actions. For each event an arbitrary number of action groups are defined. Each action group represents a number of mutually exclusive actions. The edges of the tree are attributed with selection probabilities for the respective action groups or actions.

*System Environment.*

On the system level an additional data structure (**normative board**) is necessary, which contains regular norms, valid for the complete model. Again, this can be given in advance by the modeller or derived by the evaluation of preliminary norms from the agent's normative frames. Consequently, an entry of the normative frame consists of the same elements as the normative frames, except the validation history, which is used only in the normative frame to decide if a preliminary norm will become a norm or not.

## Agent Behaviour

Basically, the agent behaviour is triggered by discrete events in the form of incoming messages. The (UML-based) activity diagram in Figure 2 shows the main loop of processing incoming messages including the pre-processing steps which are necessary for processing environmental (ENV) and norm-invocation (NI) events later on.

Thus, the first activity **"determine role"** of this process is dedicated to the distinction of the role of the message receiver **x** (actor or observer), stored temporarily in the state object R(x). Secondly, based on the role information (x) and on the content (i) of the message field "Recipient", the relevant event board of the message receiver has to be selected (**"select actual event board"**) and stored in EB(x, i). This is either the "actor" event board, or the event boards for one of the observed agents. Finally, the identification of the incoming message type is done by evaluating the modal of the message (**"check modal"**).

Modal values of A (=Assertion) and B (=Behaviour) identify "environmental" (ENV) events, whereas modal values of D(=Deontic), V(=Valuation), and S(=Sanction) characterize "norm invocation" (NI) events. Again this information is stored temporarily in M(x).

*Environmental event processing.*

Figure 3 shows the processing of incoming environmental events for agent x.

An incoming environmental message triggers this process. The message originates either from the perception of an event within the agent's environment (a so-called assertion), or is the capture of an assertion or behaviour message from an observed agent.

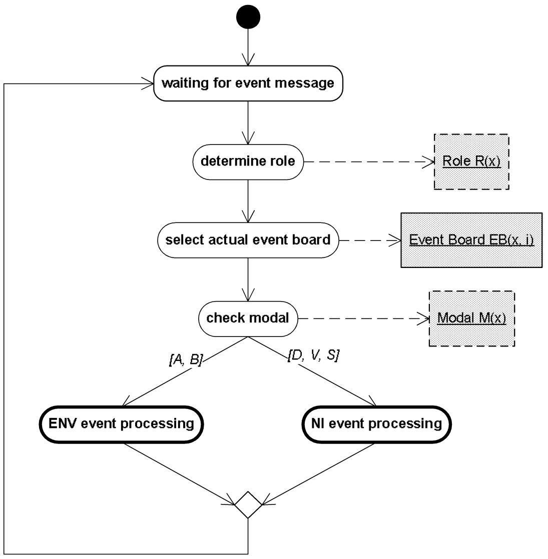


Figure 2: EMIL-S event processing

The first activity **"modify event board"** stores information about the incoming event into agent x's event board EB(x, i) for agent i. This means that EB(x, x) is the "own" actor role event board, while all other event boards are for observed agents. If the current modal is "Assertion" (M(x)=A) a new event board entry is generated, whereas the action field of an already existing event board entry is completed if the modal is "Behaviour" (M(x)=B).

The recently created or modified entry is the base for the **"calculate classifier"** activity This classifier represents the event board subsequence of a certain length (determined by a model parameter), with this entry as most recent element. The actual classifier is then saved in the temporary state object CLA(x).

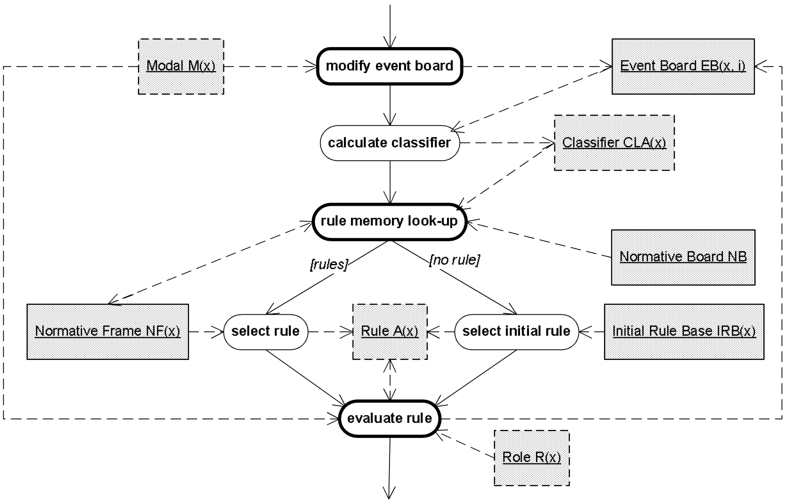


Figure 3: Environmental (ENV) event processing

CLA(x) is the key for the following **"rule memory look-up"** activity. This complex activity searches for an already existing rule for the sequence of currently recorded events in both of the long-term memories NB and NF(x). Within this activity two steps are processed:

1. The common normative board NB is examined for norms valid according to the classifier. If norms are found, they are copied into agent x's normative frame NF(x) after a decision process which reflects the agent's disposition on abiding by norms.
2. The normative frame NF(x) is searched for matching entries which can be preliminary norms or norms copied from the normative board before.

If one or more entries are found, one of them is finally selected within the activity **"select rule"** and stored in the state object A(x). This procedure is important to recognise typical and already known (complex) situations within the environment at an early stage to allow an adequate and timely reaction (e.g. to avoid undesired incidents).

In case that no normative frame entry is found, the event-action-tree that belongs to the incoming event is fetched from the initial rule base (IRB) within the activity **"select initial rule"**.

In both cases, a rule (represented by an action tree) is available as input for the activity **"evaluate rule"**. The effect of this complex activity is furthermore determined by modal M(x) and role R(x). The following modes of operation are specified:

- R(x)=ACTOR: The rule is evaluated for selecting and executing appropriate actions (by sending a message with modal B), and therefore determining the agent's behaviour. Afterwards the modified rule (in terms of reinforcing the just selected actions) is stored in the event board entry that was created at the beginning of the process.
- R(x)=OBSERVER and M(x)=A: A deontic (i.e. a norm-invocation message with modal D) is sent to the observed agent, expressing which actions would be executed by the observer agent according to its own rule.
- R(x)=OBSERVER and M(x)=B: A valuation or sanction (i.e. a norm-invocation message with modal V or S) is sent to the observed agent, blaming or praising the action which the observed agent has executed. Type and strength of the norm invocation is determined by comparing the observed actions with the rule of the observing agent.

Based on the sketched process the agent's norm oriented behaviour is implemented.

*Norm-invocation event processing.*

Incoming norm invocation events are handled by the process specified in Figure 4.

This process is triggered by the reception of messages that contain either valuations or sanctions with respect to already executed actions, or deontics disclosing information about what to do in environmental situations already experienced. In both cases, the norm invocation refers to data stored within the event board. Thus, the first activity implements an **"event board look-up"** which usually returns the valuated event entry. If no entry is found, then the valuation is invalid (e.g. the valuated action is outdated and already removed from the event board), and the processing is aborted. On the other hand, if an entry is found it becomes the most recent event of an event board subsequence for which a new classifier CLA(x) is generated within the activity **"calculate classifier"**. This activity as well as the following activity **"rule memory look-up"** are identical with the equally labelled activities of the environmental event process introduced in the previous section.

Again, the result of the look-up process is either

- a set of rules, from which the entry with the highest similarity related to the classifier is selected and stored in A(x) by **"select rule"**, or
- the information that there is no similar rule found. In

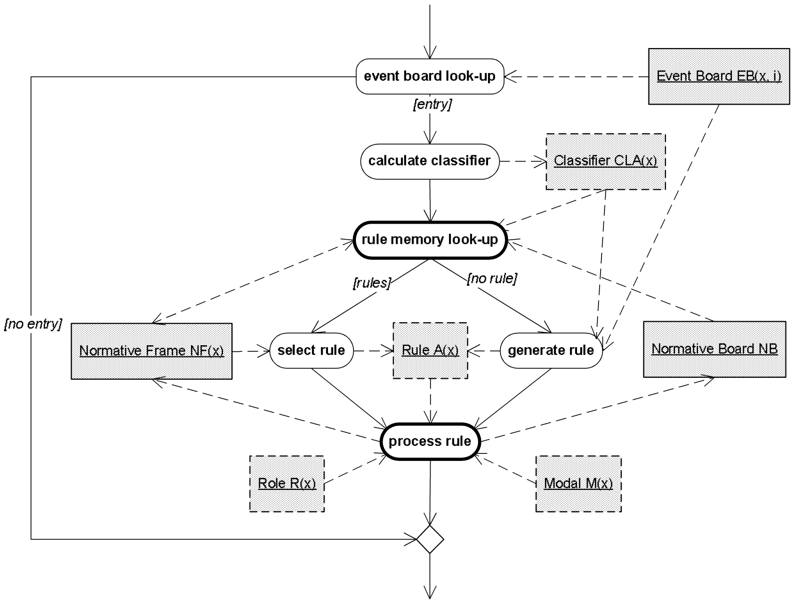


Figure 4: Norm-invocation (NI) event processing

this case, a new rule is generated by merging the rules stored in the event board subsequence for which the classifier has been calculated (**"generate rule"**).

This (either new or already existing) rule then undergoes a complex sub-process within the **"process rule"** activity, parameterised by state objects R(x) and M(x). Basically, this activity covers the following steps:

- According to R(x) and M(x), the probabilities within the rule A(x) are modified in a way that the execution of positive valuated (or sanctioned) actions will be more likely in the future, and vice-versa.
- The current norm invocation is added to the valuation history.
- The valuation history is inspected in order to decide whether the preliminary norm can be transformed into a regular norm. This decision algorithm considers (a) from how many different agents valuations are stored in the valuation history, and (b) the change rates of the probabilities attached to the action tree during the recent time period. If the agent decides to transform the preliminary norm into a regular norm, the content of the normative frame entry is proposed to the normative board NB.

This brief overview should nonetheless give an impression of the norm formation process implemented in EMIL-S.

## CONCLUSIONS

The norm formation process described in the previous section is the basis for the agent implementation in the EMIL-S simulator. EMIL-S is not realized as a stand-alone software, but rather as an extension of existing simulation scenarios (e.g. based on REPAST [North et al., 2006] or TRASS [Möhring and Lotzmann 2009]). Thus, the EMIL-S software provides an interface to concrete simulation scenarios, and an agent design user interface, which facilitates the input of event-action trees.

At present, the concept of describing norm formation models by event-action trees is tested by transforming different model scenarios into this language:

- Traffic scenario (Möhring and Lotzmann, 2009)
- Wikipedia scenario (Troitzsch, 2008)
- Microfinance scenario (Anjos, 2008)
- Scenario on conformity in multiple contexts (Andrighetto et al., 2008)

Except for the traffic scenario, these models are basically replications of existing models. It could be shown that it is easily possible to translate a broad range of widely differing models by using the concept presented.

In all cases the integration of normative observer processes constitutes a substantial model extension. The investigation of the consequences for the replicated models as well as the analysis of the (promising) results achieved so far is subject of the current work.

## REFERENCES

Andrighetto, G.; R. Conte; P. Turrini; and M. Paolucci. 2007. "Emergence In the Loop: Simulating the two way dynamics of norm innovation." In *Dagstuhl Seminar on Normative Multi-agent Systems*, Dagstuhl, Germany.

Andrighetto, G.; M. Campenni; R. Conte; and F. Cecconi. 2008. "Conformity in Multiple Contexts: Imitation vs. Norm recognition." World Congress on Social Simulation (WCSS), Faifax, July 14-17.

Anjos, P. L. dos; F. Morales; and I. Garcia. 2008. "Towards analysing social norms in microfinance groups." 8th International Conference of the International Society for Third Sector Research (ISTR), Barcelona.

Boella, G.; L.v.D. Torre; and H. Verhagen (Eds.). 2007. *Normative Muli-agent Systems.* Dagstuhl Seminar Proceedings 01722.

Bratman, M. 1987. *Intensions, Plans and Practical Reasoning.* Stanford: CSLI Publications.

Broersen, J.; M. Dastani; Z. Huang; and L.v.D. Torre. 2001. "The BOID-Architecture: Conflicts between Beliefs, Obligations, Intensions, and desires." In: *Proceddings of the 5th International Conference on autonomous agents*.

Campenni, M.; G. Andrighetto; F. Cecconi; and R. Conte. 2008. "Normal = Normative? The Role of Intelligent Agents in Norm Innovation." Fifth Conference of the European Social Simulation Association (ESSA), Brescia, Sepember 1-5.

Durkheim E. 1982 [1895]. *The rules of the sociological method.* Translated by W.D. Halls. The Free Press, New York.

Möhring, M.; Lotzmann, U. 2009. "Simulating Normative Behaviour and Norm Formation Processes." In *Proceedings of the 23nd European Conference on Modelling and Simulation*, J. Otamendi, A. Bargiela, J. L. Montes, L. M. D. Pedrera (Eds.). Madrid, June 9-12. 187-193

Neumann, M. 2008. "A Classification of normative architectures." World Congress on Social Simulation (WCSS), Fairfax, July 14-17.

North, M. J.; N. T. Collier; and J. R. Vos. 2006. "Experiences Creating Three Implementations of the Repast Agent Modeling Toolkit." In *ACM Transactions on Modeling and Computer Simulation*, 16(1), 1-25.

Troitzsch, K. G. 2008. "Simulating Collaborative Writing: Software Agents Produce a Wikipedia." Fifth Conference of the European Social Simulation Association (ESSA), Brescia, Sepember 1-5.

## AUTHOR BIOGRAPHIES

**ULF LOTZMANN** obtained his diploma degree in Computer Science from the University of Koblenz-Landau in 2006. Already during his studies he has participated in development of several simulation tools. Since 2005 he has specialized in agent-based systems in the context of social simulations and is developer of TRASS and EMIL-S. Currently he is doctoral student at the University of Koblenz-Landau.

**MICHAEL MÖHRING** received his first and his PhD degree from the computer science faculty of the University of Koblenz-Landau and has worked in the research group for about 20 years. He developed a multi-level simulation tool called MIMOSE and participated in all the research projects of the group, currently specialising in both multi-agent simulation and in data mining; he has been teaching both information society and data mining for more than a decade.

# A SYNCHRONIZATION PROTOCOL FOR DISTRIBUTED AGENT-BASED SIMULATIONS WITH CONSTRAINED OPTIMISM

Dirk Pawlaszczyk
Steffen Strassburger

School of Economic Sciences
Ilmenau University of Technology
Helmholtzplatz 3, 98693 Ilmenau, GERMANY
Email: pawlaszczyk@hotmail.com / steffen.strassburger@tu-ilmenau.de

**KEYWORDS**
Agent-based simulation, optimistic synchronization, contrained optimism, FIPA interaction procotols.

## ABSTRACT

Agent-based simulation (ABS) is a paradigm which has received much attention within the last years. For enabling industrial use of ABS scalable solutions are required which can be executed on a distributed computing architecture. Such solutions must be capable of simulating complex models with hundreds or more complex deliberative agents with really autonomous behavior.

In this article we argue that only optimistic synchronization protocols are potentially capable of providing the required performance. We suggest a synchronization protocol with constrained optimism which exploits specific characteristics of communication patterns within agent based simulations. Furthermore the presented protocol includes appropriate methods for GVT computation and fossil collection in distributed ABS as well as mechanisms to ensure repeatability.

## INTRODUCTION

In agent-based simulation (ABS) real world systems are modeled using multiple agents. The modeled system emerges by interaction of the individual agents as well as their collective behavior. Agents typically send messages with respect to some communication protocol. In this context a software agent is defined as a program that acts autonomously, communicates with other agents, is goal-oriented (pro-active) and uses explicit knowledge.

Agent-based modeling and therefore agent-based simulation seems to be an appropriate tool for domains characterized by discrete decisions and distributed local decision makers. With growing complexity of agent based models, the scalability of a simulation environment becomes a crucial measure. To simulate an increasing number of entities, the underlying simulation system needs to be scalable, thus creating an immediate demand for distributed simulation.

Although ABS has received a lot of attention in recent years there are rather few contributions in place that deal with the problem of scalable distributed agent-based simulation. The objective of the research presented here is to discuss in detail an approach for a scalable time synchronization algorithm which takes advantage of the specifics of the agent-based simulation approach effectively.

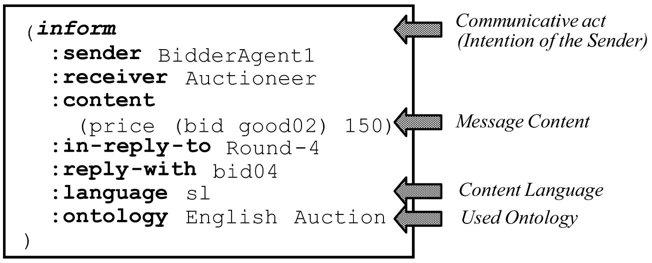
## BASICS OF AGENT-BASED SIMULATION

### Agent Technology and Standards

When using agent-based design approaches it is often the interplay between multiple agents, which one is interested in. This leads to the term of multi-agent systems (MAS). An MAS is generally considered a system composed of multiple autonomous agents which can interact with each other. Multi-agent systems can be used to solve or describe problems which are difficult or impossible to solve/describe with individual agents or a monolithic system.

It is often assumed that agents in MAS are executed and interact with each other in real-time, i.e., there is typically no separate logical time representation within an agent as it is known from paradigms like discrete event simulation. Please note that this is often different when talking about ABS, as discussed further down in the paper.

Agents in MAS are only capable of exchanging knowledge and interacting when they use a common language understood by all agents. The Foundation for Intelligent Physical Agents (FIPA) is an organisation founded with the objective of creating a framework architecture for the interaction of heterogeneous agent systems. A central point of this effort are standards for message-based communication of agents and multi-agent systems. The structure of a message is defined by the Agent Communication Language (ACL) (FIPA 2002). Within messages the communicative act, i.e. the intention of the sender, is identified using performatives like “inform”, “request”, “agree”, etc. Figure 1 gives an ACL message example which is part of an auction protocol.



```
(inform
  :sender BidderAgent1
  :receiver Auctioneer
  :content
    (price (bid good02) 150)
  :in-reply-to Round-4
  :reply-with bid04
  :language sl
  :ontology English Auction
)
```

**Figure 1: Example for an ACL-Message**

Most importantly for the further discussions, FIPA also defines specifications for interaction protocols (FIPA 2002a). These interaction protocols define typical sequences of messages or patters, how agents may interact. The resulting dialogs between agents always follow this same pattern. A simple example of this is shown in Figure 2.

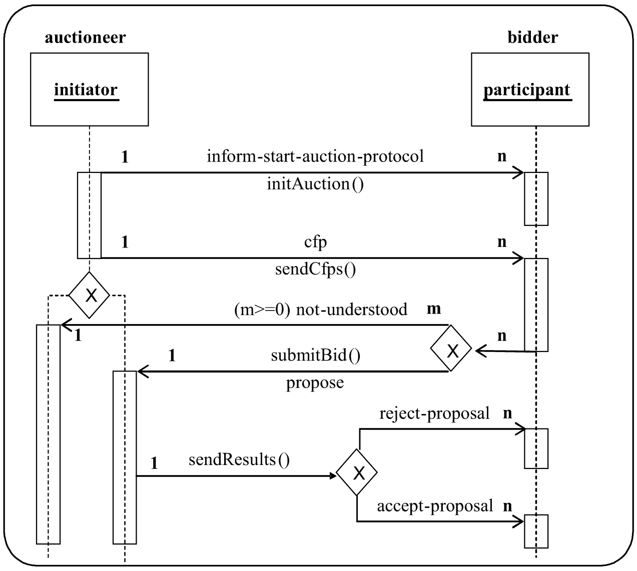


**Figure 2: Example for an auction protocol**

The figure depicts the possible lines of communication between an auctioneer agent and multiple bidder agents. The auctioneer first has to request bids from the participating bidders, before he can accept a proposal and inform the successful bidder.

The possible types of the messages exchanged within this interaction protocol and their sequence is independent from the actual message content. It is, for instance, completely irrelevant whether the auction concerns a pencil or a car.

The most frequently used interaction protocols between agents are firmly defined within the described FIPA standards (compare Table 1).

**Table 1: Some FIPA Interaction Protocols (FIPA 2002a)**

| Title | Description |
|---|---|
| FIPA Request Interaction Protocol Specification | Allows one agent to request another to perform some action. |
| FIPA Query Interaction Protocol Specification | Allows one agent to request to perform some kind of action on another agent. |
| FIPA Request When Interaction Protocol Specification | Allows an agent to request that the receiver perform some action when a given precondition becomes true. |
| FIPA Contract Net Interaction Protocol Specification | Allows one agent (the Initiator) to take the role of a manager which wishes to have some task performed by one or more other agents (the Participants) and further wishes to optimize a function (e.g., price) that characterizes the task. For a given task, any number of the Participants may respond with a proposal; the rest must refuse. Negotiations then continue with the proposer. |
| FIPA Propose Interaction Protocol Specification | Allows an agent to propose to receiving agents that the initiator will do the actions described in the propose communicative act when the receiving agent accepts the proposal. |

The interaction protocol to which a message belongs is contained within a message field. By using a certain protocol the agent commits to react to requests of the protocol in the predefined format.

The FIPA Interaction Protocols are designed for usage in MAS in general. In our work, we adopt them to be used in ABS and we take advantage of the fact that they define commonly used interaction *sequences*.

### Agent-based Simulation (ABS)

The term agent-based simulation (ABS) describes the modeling and simulation of real systems with the help of agents, which interact within a simulation model.

The usage of agent based approaches for modeling and simulation promises greater flexibility and better abstraction capabilities when describing the behavior of systems with many active (or "intelligent") components. In agent-based simulations agents can represent workers, machines, carriers, etc. which can all have their autonomous behaviour.

There is a wide variety of development environments for agent-based simulations, mostly from academic sources. Current examples include Cougar, Farm, James II, Repast, Samas, Sassy, and many more.

Please note that most of these environments use event based mechanisms for advancing logical simulation time. This shows the strong influence that paradigms like discrete event simulation have had on ABS. In fact, one could argue that ABS as a modeling philosophy is quite similar to modeling with object-oriented simulation tools with process-oriented world views like SLX (Henriksen 1997).

In the further discussion we limit the scope of our work to this type of ABS. We do not consider ABS which operate in a real-time manner like common MAS.

### Distributed Simulation and ABS

According to Fujimoto (2000) distributed simulation (DS) is a technology that enables a simulation program to be executed on distributed computer systems.

Agent-based modeling and simulation environments do not necessarily have to be build in a distributed fashion (Uhrmacher and Gugler 2000). However, when scalability issues have to be considered, the parallel or distributed execution of agents is the only technology offering hope for increased performance when the problem size (i.e. the number of agents and their complexity) increases beyond that what a single machine can simulate or process. Main motivation for suggesting the usage of distributed simulation for agent-based models is therefore the scalability aspect.

For enabling scalable agent based simulation, both hardware and software (i.e. the applied algorithms) has to be scalable. The focus of the discussion in this paper is on the software aspect, specifically on a synchronization algorithm that scales well for agent based models and their specific characteristics.

## OPTIMISTIC SYNCHRONISATION OF DISTRIBUTED AGENT-BASED SIMULATIONS

Synchronization protocols as the core technology needed for distributed simulation can be classified into the two main categories of conservative and optimistic protocols (Fujimoto 2000).

Conservative protocols implement mechanisms that prevent a member of a distributed simulation from processing messages out of time stamp order, thus maintaining strict causality. Conservative synchronization protocols are rather easy to use and implement, but their performance depends highly on a value called Lookahead. Lookahead is a guarantee from a simulation that it will not generate any messages with a time stamp smaller than its current time plus the value of Lookahead. If a simulations' current time is T, and its Lookahead is L, any message generated by the federate must have a time stamp of at least T+L.

Lookahead is hard to extract and always depends on the application context. For agent-based simulations, the situation comes close to the worst case scenario, as their interaction protocols often require immediate answers resulting in a Lookahead requirement. In this case, conservative synchronization protocols almost completely inhibit parallelism within the distributed simulation.

Optimistic protocols do not impose the requirement to process events in strict time stamp order. Simulations using optimistic synchronization can process received messages although there maybe future messages with a smaller time stamp. To maintain causality, these approaches detect and recover from causality errors, which may be introduced by processing events before it is safe to proceed. The major advantage of these approaches is that they allow the exploitation of parallelism in situations where it is possible that causality errors might occur, but in fact they do not occur. The Time Warp protocol is an example of an optimistic synchronization mechanism.

## Experimentation Framework

Optimistic protocols are more complex to implement than conservative protocols, but the following statement of Fujimoto (2000) certainly holds much truth: "If one's goal is to develop a general purpose simulation executive that provides robust performance across a wide range of models […] optimistic synchronization offers greater hope." Consequently, our research has focused on this approach and has tried to intelligently combine it with ABS in order to eliminate some of the problems which are inherent to optimistic synchronization techniques.

We have developed a simulation kernel to enable efficient simulation of large-scale agent based models. Therefore, we have added parallel discrete event simulation (PDES) functionalities on top of an existing agent middleware to get support for optimistic simulation.

Our implementation is based on the Java Agent Development Environment (JADE), a generic framework for development of agent based applications (JADE 09). JADE offers an appropriate middleware to simplify the implementation of multi agent systems. It is widely used in academia. Since JADE is compliant to the FIPA standard, a high degree of interoperability is guaranteed. Moreover, the JADE messaging sub system scales well, even for heaviest message traffic (Vitaglione et. al 2002). JADE provides many built in features like remote method invocation (RMI), serialization, and agent management tools that allow a distributed model to be easily developed. Because the program is written in the Java programming language the implementation is portable across a wide range of platforms.

Applying conventional PDES models and techniques to MAS is more complicated than one might think, since we have to regard specifics of agent technology like a particularly high communication demand and dynamic topologies, without degrading performance. At the heart of the simulation executive there is a new optimistic synchronization algorithm. Furthermore we have implemented an efficient decentralized scalable algorithm for computation of GVT (global virtual time) taking into account transient messages without using blocking barriers. Other basic features of the simulator include automatic state saving, repeatability of simulation runs using a tie-breaking algorithm (Mehl 1992) and simulation support for FIPA compliant agent models. All PDES-functionalities were added to the JADE middleware as platform services using the standard plug-in concept of this agent-framework. This enabled us to test the feasibility of the newly developed algorithms.

Programming simulation models with the simulator is straightforward. Individual agents are programmed by the application developer using the standard agent-based sense-think-act paradigm. Agent processes are autonomous in the sense that they hold and manage their own events, and are optimistic in their event processing. State saving and synchronization of agent processes is done in background.

## Optimistic Processing

A scalable architecture by default requires it to be composed of multiple processors/computers, i.e., so that the number of used resources can grow as the problem size increases. Searching for a synchronization technique we have disqualified conservative approaches because of the inherent zero lookahead requirements in many agent interaction protocols. Hence, optimistic synchronization seems to be the most promising approach, because it (at least theoretically) provides enough performance for a wide range of models.

However, within pure Time Warp implementations, a common problem can be caused by rollback cascades as there are no constraints on the relative distance between logical processes (LPs). One can call this a problem of "too much optimism". Every LP processes events almost independently of the progress of other LPs' timelines. Consequently, the probability of incorrect computations (i.e. causality errors) is very high. If the time to perform a rollback is high, i.e. many states have to be rolled back, the performance of the simulation rapidly decreases.

A good time management algorithms therefore has to avoid situations like these while still ensuring a high degree of parallel execution.

The basic idea of our suggested approach is very straight-forward: We suggest to limit the level of optimism in the applied time warp protocol by using extra-knowledge extracted from the used agent interaction protocols. Communication following these interaction protocols is one of the key features in agent technology.

Messages are sent out from a sender to one or more receiver(s). Messages are encoded in an *Agent Communication Language (ACL)*, an external language that defines the intended meaning of a message by using performatives. A series of messages produces a dialog. A dialog normally follows a predefined structure – the *Interaction Protocol* (IP). The FIPA *Request Interaction Protocol* for example allows an agent to request another agent to perform some action. The responding side needs to decide whether to accept or refuse the request. In any case, the message receiver has to respond. Even if the receiver cannot interpret a message, the specification prescribes to send a least a not-understood message.

We exploit this characteristic: the communication almost always follows a known sequence of messages. In normal Time Warp every new event is immediately sent over the network to its corresponding receiver where it can be executed immediately. Our approach leverages delays on the event execution based on information about the current state of the interaction protocol. Instead of immediately processing every incoming event message the event is delayed using an adaptive rule:

**Definition 1** (wait for rule): Given agent $a_1$ which has sent a message $m_1$ to agent $a_2$, and assuming that there is at least one valid required reply $m_2$ for $m_1$, the rule "wait for" is defined as follows: If the expected reply message was not received yet, the agent $a_1$ must wait for this particular message, before going on to process the next message.

This rule basically requires agents to wait for a reply if a communication which is part of an interaction protocol has been started. In the *FIPA-request-protocol* for example, if an agent has agreed to do something for its opponent he automatically commits himself to send an *inform-done-message* as soon as he has finished the task. If he did not succeed he has to send a *refuse-message*. In any case the agent always has to reply. Accordingly, for every message $m_i$ which is received while agent a is waiting for message $m_k$ from a sender different to the sender of $m_i$, this message is buffered. The execution of $m_i$ is delayed.

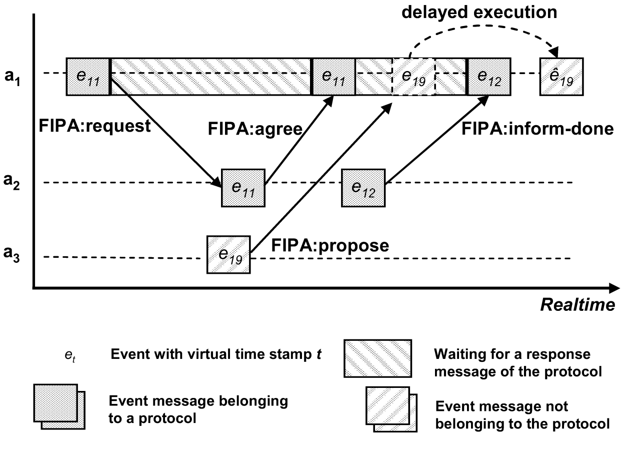

delayed execution
a1
a2
a3
e11
e11
e19
e12
ê19
e11
e12
e19
FIPA:request
FIPA:agree
FIPA:inform-done
FIPA:propose
Realtime
e_t Event with virtual time stamp t
Waiting for a response message of the protocol
Event message belonging to a protocol
Event message not belonging to the protocol

**Figure 3: Delayed event execution based on protocol information. Agent $a_1$ receives a proposal from Agent $a_3$, while he is waiting for an inform-done message of Agent $a_2$. Instead of immediately processing the incoming message, the execution is delayed. Thus, event order is preserved and still valid**

In Figure 3 an example is given to demonstrate the effect of delayed execution. The events in this example have distinguishable time stamps only for reasons of clarity. The suggested synchronization protocol can handle simultaneous events just in the same way.

In the example, the immediate execution of event $e_{19}$ from $a_3$ normally would cause the agent process $a_1$ to rollback when message $e_{12}$ is received at some later point in time. With the new policy in place this situation can easily be avoided. Instead of immediately processing message $e_{19}$, agent $a_1$ has to wait for the reply message from agent $a_2$. This is because there is an external knowledge that the active interaction protocol will sooner or later require $a_2$ to send a message "FIPA: inform-done" (expressed as $e_{12}$). The wait-for-rule basically formalizes this behaviour. Therefore event $e_{19}$ has to be buffered thus preserving the relative event order. This approach minimizes the potential for incorrect execution of events as far as knowledge from the interaction protocols can be extracted.

The process of waiting for a certain message could be interpreted as a conflict to the asynchronous nature of agent execution, but in fact, it is not a contradiction, rather, it is a natural approach common in PDES to maintain causality. Agents in our simulations must behave consistently with the interaction protocol specification. If agent $a_1$ in the example above executed $e_{19}$ immediately upon reception, a rollback would be unavoidable when $e_{12}$ is received. Since the agent has the external knowledge that $e_{12}$ will in fact occur, it is straight-forward to avoid this situation. It should also be noted, that the result of the simulation is not affected in any way by this approach.

Further to the basic example given above, the implementation of the presented approach has to take into account some special cases where exceptions to this rule may be needed. First, let us have a look at the general decision tree which is passed whenever a new message is sent from one agent to another (Figure 4).

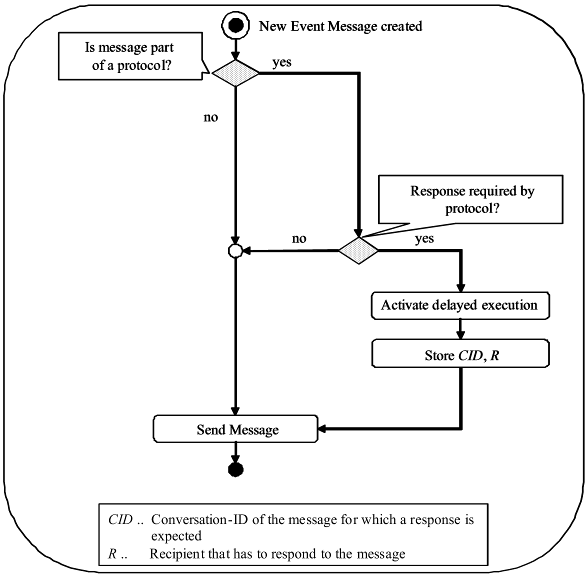

New Event Message created
Is message part of a protocol?
yes
no
Response required by protocol?
no
yes
Activate delayed execution
Store CID, R
Send Message
CID .. Conversation-ID of the message for which a response is expected
R .. Recipient that has to respond to the message

**Figure 4: Algorithm for sending new messages and enabling constrained optimism**

If the message is part of an interaction protocol and a response is required by this protocol, the delayed execution is activated and the message is flagged. All further received messages are subject to the wait for rule and their execution is potentially delayed.

However, there may be some situations where exceptions to the wait-for rule may be required. In certain situations an agent must exceptionally be allowed to execute events, even if it is waiting for a reply message. Consider, for example, the following potential *deadlock* situations. Assuming an agent $a_1$ waits for agent $a_2$, and at the same time agent $a_2$ waits for agent $a_1$. This may be the case since both independently have sent a message to each other, each being part of a different conversation. Both agents would then become blocked as they are both waiting for a event message which will never occur. In this case an agent must be allowed to process a message from its opponent even if it is not the message content it was waiting for. Another exception are *cyclic dependencies*. Although not very likely, there may be situations when an agent receives a request within the same conversation, from a new communication partner different from its original opponent. This may be the case for example in multi-staged-protocols. In this situation, this new message has to be processed by the waiting agent before he can go to wait state again.

Figure 5 illustrates the decision tree at the receiver side which takes these special situations into account and bypasses the delayed execution if needed.

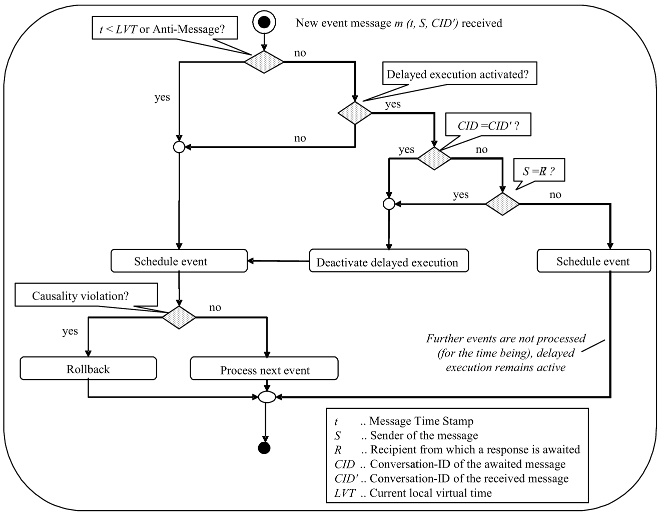

t < LVT or Anti-Message?
New event message m (t, S, CID') received
no
yes
Delayed execution activated?
yes
no
CID =CID' ?
yes
no
S =R ?
yes
no
Schedule event
Deactivate delayed execution
Schedule event
Causality violation?
no
yes
Rollback
Process next event
Further events are not processed (for the time being), delayed execution remains active
t .. Message Time Stamp
S .. Sender of the message
R .. Recipient from which a response is awaited
CID .. Conversation-ID of the awaited message
CID' .. Conversation-ID of the received message
LVT .. Current local virtual time

**Figure 5: Algorithm for processing received event messages taking into account necessary exceptions**

The so described synchronization algorithm is joining optimistic techniques with constrained optimism and can therefore be classified as a time warp with constraints. The proposed policy certainly cannot fully prevent rollback situations. However, it minimizes the risk for rollbacks caused by messages which causally belong to a common communication. For agent-based simulations this approach, according to our experiments, performs better than a pure Time Warp solution. Also, this approach is capable of maintaining an acceptable degree of parallelism required for scalable solutions.

The implementation effort of this solution is considerably low. The agent has to be provided with information about the structure of the used protocols at initialization time only. Depending on the protocol length, the policy is applied more frequently. Particular long interaction protocols, like the *fipa-contract-net* are most eligible.

## SUMMARY AND CONCLUSIONS

While agent-based simulation as a methodology has received much attention from the research community in the past, scalability and thus industrial applicability of this technology has been somewhat neglected. In this paper we have argued that only the efficient distributed simulation of agent-based models can provide the architectural basis for a scalable solution. We have further on argued that only the deployment of an optimistic synchronization protocol can yield the required performance. Further on, we have in detail presented a constrained optimistic synchronization protocol, which eliminates a significant amount of the drawbacks and risks that exist within the original optimistic time warp protocol.

This is achieved by taking advantage of specific characteristics that are inherent in many agent based interaction protocols making it therefore a good solution for scalable distributed simulation of agent based models which are based on logical simulation time. Empirical performance results for a variety of tests support this statement (Pawlaszczyk and Strassburger 2009).

## REFERENCES

FIPA. 2002. FIPA ACL Message Structure Specification. FIPA Spec 00061. Available via http://www.fipa.org/specs/fipa00061/SC00061G.pdf [accessed April 1, 2009].

FIPA. 2002a. FIPA Interaction Protocols. FIPA Specs 0026-0036. Available via http://www.fipa.org/repository/standardspecs.html [accessed April 1, 2009].

Fujimoto, R. 2000. *Parallel and Distributed Simulation Systems*. Wiley Interscience, 2000.

Henriksen, J.O. 1997. An introduction to SLX. In *Proceedings of the 1997 Winter Simulation Conference*, ed. Andradóttir, S., K. Healy, D. Withers, and B. Nelson, 559-566. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

JADE 2009. Homepage ot the JADE project: http://jade.tilab.com [accessed April 1, 2009].

Pawlaszczyk, D., S. Strassburger. 2009. Scalability in Distributed Simulations of Agent-Based Models. To appear in: *Proceedings of the 2009 Winter Simulation Conference,* eds. M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls. December 13-16, 2009. Austin, USA.

Mehl, H. 1992. A Deterministic Tie-Breaking Scheme for Sequential and Distributed Simulation. In: *Proceedings of 6th the Workshop on Parallel and Distributed Simulation*, ed. M. Abrams and P. F. Reynolds, 199-200.

Uhrmacher, A. M., K. Gugler. 2000. Distributed, parallel simulation of multiple, deliberative agents. In: *Proceedings of the 14th Workshop on Parallel and Distributed Simulation*, 101-108. Washington, DC, USA: IEEE Computer Society, 2000.

Vitaglione, G., F. Quarta, E. Cortese. 2002. Scalability and Performance of JADE Message Transport System. In: *Proceedings of the AAMAS Workshop on AgentCities*, Bologna, 2002.

# WATER MANAGEMENT SYSTEMS

# SOME ALGORITHMS SUPPORTING THE COMPUTER AIDED MANAGEMENT OF COMMUNAL WATER NETS

Jan Studzinski
Polish Academy of Sciences, Systems Research Institute
Newelska 6, 01-447 Warsaw, Poland
E-mail: studzins@ibspan.waw.pl

**KEYWORDS:** Waternet hydraulic models, multicriterial optimization, model calibration, kriging algorithms, monitoring systems, decisions support systems.

## ABSTRACT

An information system designing for the complex management of a municipal water net consists of several modules which are destined for solving of many operational tasks like digital visualization of the water net, its mathematical modeling, optimization, control and monitoring, registration of the water net customers etc. These modules consists mostly of several programs including the algorithms whose role is to solve many fragmentary problems of the water net management. These algorithms called tooll algorithms are not essential in the construction of the information system but without them the system efficiency would be radical worse than it could be with them. In the paper some algorithms of these accessory functions, like the automatic calibration of the water net hydraulic model, defining altitude cooordinates of the water net nodes, drawing maps for the water flow and pressure distributions and choosing the optimal locations for the SCADA measurement points are presented.

## INTRODUCTION

At the Systems Research Institute of Polish Academy of Sciences (IBS PAN) for some years an information system called MACOW (*Management And Control Of Waternets*) for the management of municipal water networks is under development (Studzinski, 2007). The system consists of four main modules which are the GIS (*Geographical Information System*) system called G/Water and developed by Intergraph Warsaw, the SCADA (*Supervisory Control And Data Acquisition*) system based on the visualization program Procon made by Schulz-Infoprod Poznan, a CIS (Customary *Information System*) system developed by a local Polish firm and a hydraulic waternet model called MOSUW and developed at IBS PAN; the model is coupled with a multicriterial optimization method developed by *Straubel* (Straubel, 2007). These modules are prepared for the self-contained as well as for cooperated operation but in the latter case some additional algorithms have to be made and included into the MACOW structure. The algorithms make the common operation of individual modules possible and because of that they extend essentially the range of tasks that can be realized by the whole information system. In the following a couple of such algorithms with the tasks they help to exercise will be presented. The information system has been tested at the waterworks of middle size in the Polish city Rzeszow. The city has got ca. 170.000 inhabitns and its waternet has the whole length of ca. 700 km with ca. 20.000 nodes and 20.000 pipes. The nominal daily capacity of the waternet is about 70.000 m$^3$ water.

## SHORT DESRIPTION OF THE MACOW MODULES

***The first*** of the main MACOW modules is the G/Water system generating the numerical map of the waternet (see Fig. 1). This is made with the help of data defined in the Branch Data Base of G/Water that collects all technical, technological and economical information concerning the waternet.

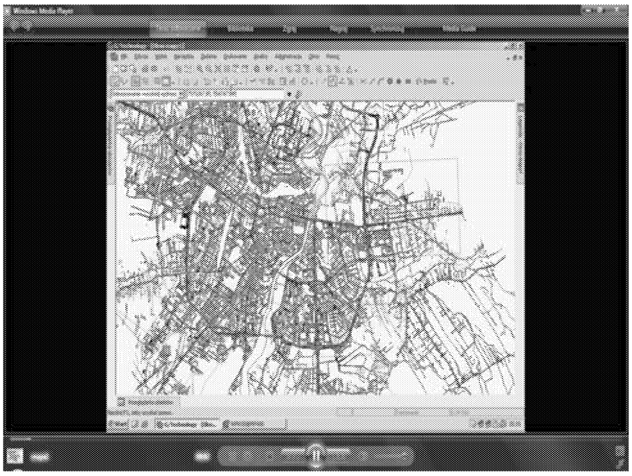


Figure 1. Numerical map of the Rzeszow waternet generated by G/Water.

***The second*** module of MACOW is the monitoring system consisting of three elements which are the measurements points located on the waternet, the system of data transmission and the program for collecting, processing and visualization of the measurements data. On the Rzeszow waternet there are 30 measurement points recording the current values of water flows and pressures and transferring them to the central unit of the SCADA system. The transfer data occurs by means of the mobile telephony using the GPRS (*General Packet Radio Service*) system and also for some measurements points by the radio connection. The net of the measurement points and the data collected there are illustrated on the computer screen by the Procon Program that performs also their assembling and processing (see Fig. 2).

***The third*** module of MACOW is the CIS system responsible for collecting the data of the waternet end-users. In this way in the data base of this system the information concerning the amounts of water drawn out in individual waternet nodes in defined time periods are accumulated.

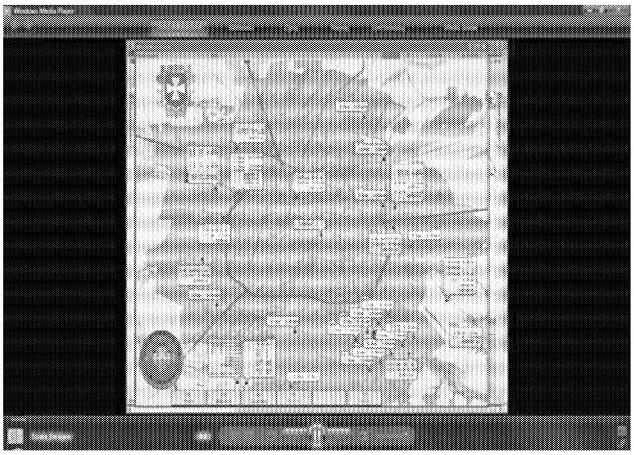


Figure 2. The monitoring system of the Rzeszow waternet visualized by Procon program.

***The fourth*** module of the MACOW system is the hydraulic model of the waternet (MOSUW) coupled with an optimization program. With the model the values of water pressures in the waternet nodes and of water flows in the waternet pipes can be calculated. The model equations are formulated using the *Kirchhoff's* laws known from the electrical engineering and using the *Bernoullie's* principle known from the flow mechanics. With the I *Kirchhoff's* law the model equations for the flow balances in all waternet nodes, linear regarding the flows $q_{ki}$ which enter and leave the node $k$ given, are formulated:

$$\sum_i q_{ki} = 0 \qquad (1)$$

With the II *Kirchhoff's* law the equations for the pressure balances in all waternet loops, nonlinear regarding the flows coursing in the loop $l$ given and linear regarding the pressure losses $h_{li}$ along the pipes forming the loop, are formulated:

$$\sum_i h_{li} = 0 \qquad (2)$$

From the *Bernoullie's* principle the linear equations for the pressure values $P_k$ in all waternet nodes result:

$$P_k = P_{k\text{-}1} - h_{ki} \qquad (3)$$

With (1), (2) and (3) the whole set of the model equations is given. The numerical computing of it occurs in MOSUW using the standard *Newton-Raphson's* method for iterative solving nonlinear algebraic equations.

The pressure losses $h_{li}$ along the pipes in the waternet are calculated using the *Darcy-Weisbach's* formula:

$$h_t = \frac{\lambda l v^2}{2gD} \qquad (4)$$

with $v = \dfrac{4q}{\pi D^2}$ and with the friction coefficient $\lambda$ described by the following *Nikuradse's* formula:

$$\lambda = \frac{1}{(2 \log \dfrac{D}{k} + 1{,}14\ )^2} \qquad (5)$$

The friction $\lambda$ depends on the pipe diameter and on the roughness coefficient $k$ whose values are given in special tables relating to the materials of the waternet pipes.

The screens of the MOSUW program are shown in Figures 3 and 4. After the model computation the calculated values of pressures and flows are marked on the waternet graph with different colors for small, high and medium values.

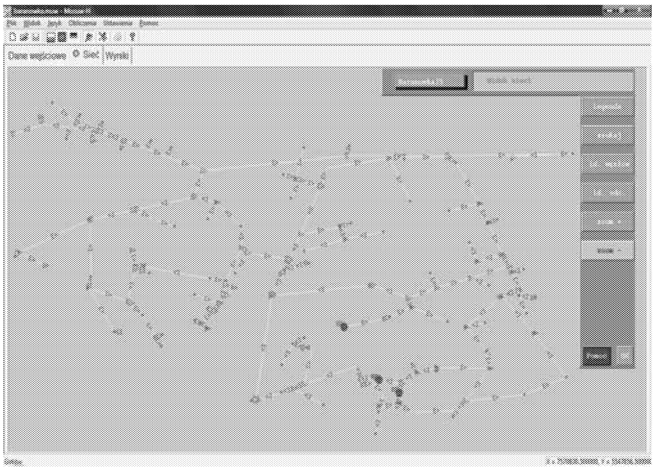


Figure 3: The screen of MOSUW with a part of the Rzeszow waternet, before the hydraulic calculation .

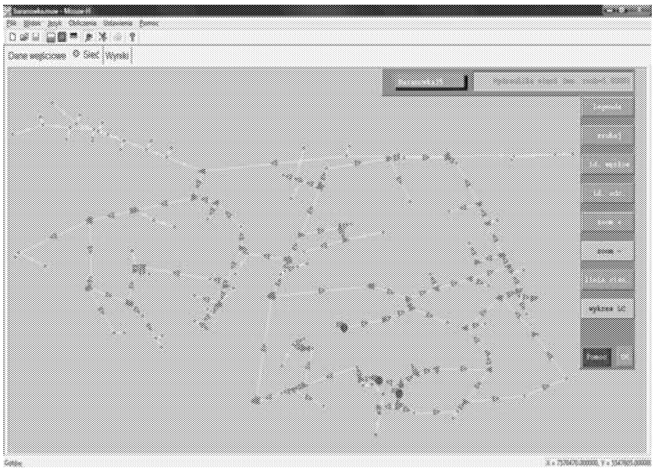


Figure 4: The screen of MOSUW after the hydraulic calculation of the waternet.

With MOSUW an algorithm for multicriterial optimization is connected. In this algorithm several goal functions can be defined and while optimizing a waternet they can be as follows:

- $F(1)$ (min): maximal difference between the given and calculated pressure values in the end-user nodes;
- $F(2)$ (min): sum of the pressure losses in all waternet pipes;
- $F(3)$ (min): maximal pumping pressure defined for the waternet pump stations;
- $F(4)$ (max): minimal velocity of the water flow in the waternet pipes;

- $F(5)$ (min): entire investment costs as a result of the waternet optimization;
- $F(6)$ (min): price of 1 m$^3$ of water.

As a result of the algorithm's execution a set of *Pareto-optimal* solutions is received from which the waternet operator can make the choice of the best one in his opinion.

## THE TOOL ALGORITHMS OF MACOW

***Buffer files***: The main modules of MACOW can operate separately but if one wants to get them cooperating then some buffer files must be used. With these files the communication occurs and the input and output data are transferring between the modules. The clue element of the whole information system is the Branch Data Base of G/Water and all intermodular communication occurs via this Base. In this way all modules are connected each other using the data files rotating between G/Water and other programs. In Fig. 5 one can see how a waternet graph designed on the numerical map (the picture on the top-right side) is transferred with a buffer file to the hydraulic model.

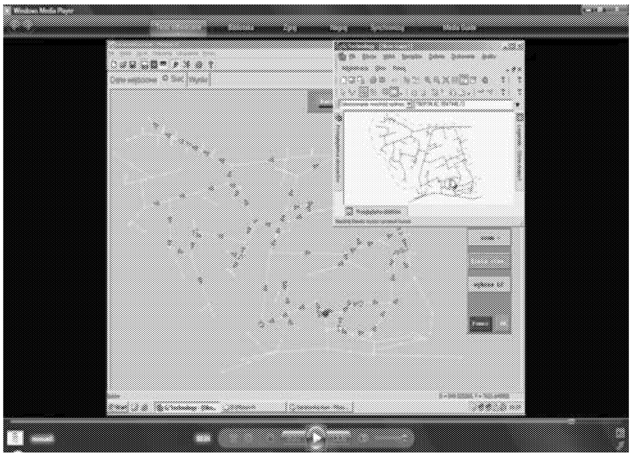


Figure 5: The connection realized between the G/Water and MOSUW.

***Calculation of the node altitudes***: The G/Water system while generating the waternet numerical map uses the data coming from the geodetic map of the waternet. But such the digital reflection of the geodetic map has got a lot of shortages which must be completed to get the waternet graph adapted for hydraulic calculation. These shortages are: many discontinuities at the waternet graph, the lack of waternet nodes which are absent on the geodetic map, and the resulted lack of the node altitudes. The complements of this deficient graph consist in elimination of these discontinuities, in creation of a new layer in G/Water containing the nodes as the new waternet objects, and in calculation of the node altitudes. For the latter task two algorithms have been developed and included into the MACOW structure. They use for their calculations the coordinates of the municipal geodetic points which are determined for each city. In Rzeszow there are 120.000 of the geodetic points fixed. The first algorithm creates a grid of triangles defined on the geodetic points (see Fig. 6). Knowing all the triangle equations one can approximate the altitude coordinate of each node locating in the relevant triangle.

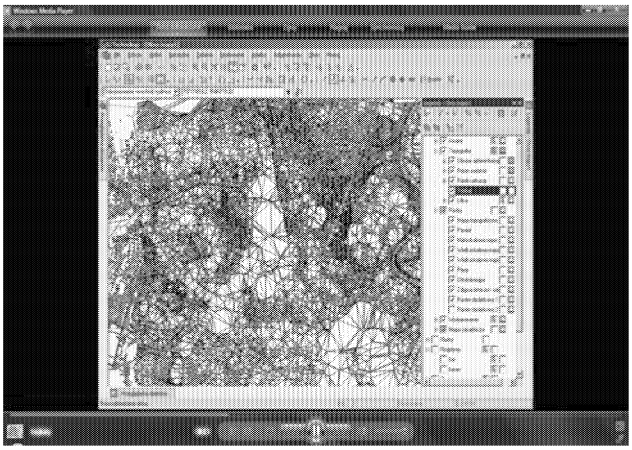


Figure 6: The grid of triangles covering the waternet area.

The second algorithm uses the kriging approximation to estimate the unknown altitude coordinates (Bogdan and Studzinki, 2007). The kriging approximation means in general the estimation of unknown values of a variable (node altitude in our case) in some selected points of an area (node points) on the base of the known values of this variable defined in other area points (geodetic points). The unknown altitude in the node point $x_o$ is estimated by the formula:

$$z(x_o) = \sum_{i=1}^{N} \lambda_i z(x_i) \qquad (6)$$

where $z(x_i)$ are the defined altitudes in $N$ geodetic points and $\lambda_i$ are some weights coefficients that must be calculated. The kriging approximation algorithm consists of 4 following steps:

*Step* 1: Calculation of the experimental semivariogram function using the altitudes of the geodetic points:

$$\gamma(h) = \frac{1}{2n_h} \sum_{i=1}^{n_h} (z_{h+1} - z_i)^2 \qquad (7)$$

where $z_i$, $z_{i+h}$ are the defined altitudes in the points outlying up to distance $h$ from each other and $n_h$ are the numbers of the pairs of such the points determined for different $h$.

*Step* 2: Modeling the experimental semivariogram using an analytical function.

*Step* 3: Calculation of the weight coefficients on the base of the conditions of unbiaseness ($E(z_i - m_z) = 0$) and of maximal effectiveness ($E[(z_i - m_z)^2] = \min$) of the altitude estimator.

As the result of these conditions the equations system results:

$$\gamma(x_j, x_0) = \sum_{i=1}^{N} \lambda_i \gamma(x_i, x_0) + \mu \qquad (8)$$

for $j = 1, 2, \ldots, N$, from which the weight coefficients can be calculated.

*Step* 4: Calculation of the altitude value in the node point $z_0$:

$$z_o = \sum_{i=1}^{N} \lambda_i z_i \tag{9}$$

The both algorithms are quite different but the differences in their results are quite the same and not bigger than a few percents (see Fig. 7).

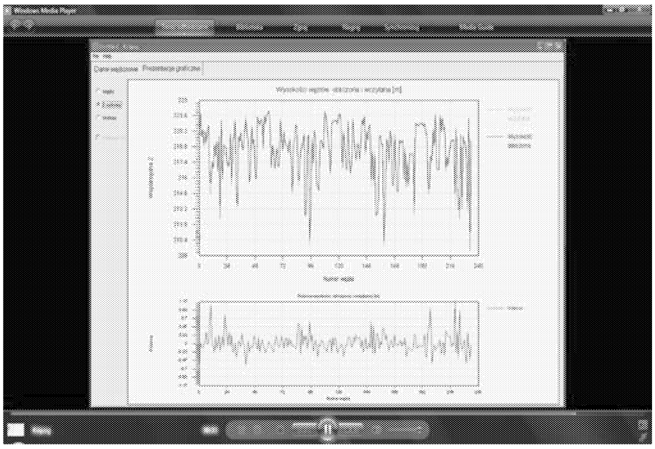


Figure 7: Approximation results (*upper diagram*) and the result differences (*lower diagram*) for both algorithms of the altitude calculation.

***Determining the measurement points for SCADA system***: The main tasks of SCADA are to give to the waternet operator the current information about the networkt work condition and to support the calibration process of the waternet hydraulic model. To do this successfully the monitoring system installed on the waternet shall have the measurement points located in sensitive network places. As a result the small number of the measurement points can supply the operator with possibly large amount of useful information. The algorithm of determining the sensitive monitoring points consists in simulation of water leaks in the successive network nodes and in calculation of the network sensitivity in the remaining nodes. The waternet sensitivity is calculated with the following relations (Straubel and Holznagel, 1998) regarding the water flow and pressure values which are computed with the hydraulic model MOSUW:

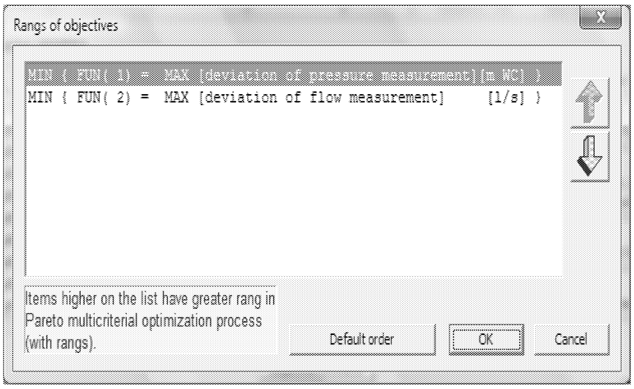
$$S_{pm} = \frac{\sum\limits_{k \neq m} (\Delta p_m / p_m) L_{km}}{\sum\limits_{k \neq m} L_{km}} \tag{10}$$

$$S_{qm} = \frac{\sum\limits_{k \neq m} (\Delta q_m / q_m) L_{km}}{\sum\limits_{k \neq m} L_{km}} \tag{11}$$

where: $k$ – the successive node with the water leak simulated, $m$ – the measurement point considered, $p$ – water pressure, $q$ –water flow, $\Delta p_m$ and $\Delta q_m$ – the differences in measurements by normal and crash cases, $L$ – the distance between the points $k$ and $m$.

The correct measurement points are these ones with the highest sensitivity values. One can see that the accuracy of the waternet model decides about the correctness of defining the well situated monitoring points.

***Hydraulic model calibration***: The accuracy of the waternet model depends on the quality of its calibration. The usual practice of the calibration is to repeat the simulation runs of the model and to change *per hand* the roughness values of the waternet pipes (see (4) and (5)) until the correspondence between the measured and calculated values of the water pressures and flows is received. The roughness coefficients are given in special tables depending on the pipe material but in the real waternets the pipe roughness is changing with the time and these changed values must be estimated. The algorithm of the waternet model calibration used in the MACOW system consists in fitting the model to the waternet not by using the *per hand* procedure but by means of the multicriterial optimization algorithm attached to MOSUW (Studzinski, 2009). To do this the waternet pipes for changing their roughness are to be fixed and the intervals of the roughness changeability for each pipe must be given. The second step of the calibration is the formulation of two criterial functions that will be optimized (see Fig. 8). These functions describe the deviations between the water pressures and the water flows calculated by MOSUW and measured at the measurement points of the SCADA system.



Rangs of objectives

```
MIN { FUN( 1) = MAX [deviation of pressure measurement][m WC] }
MIN { FUN( 2) = MAX [deviation of flow measurement]     [1/s] }
```

Items higher on the list have greater rang in Pareto multicriterial optimization process (with rangs).

Figure 8. The citerial functions defined for the calibration of the waternet model.

***Drawing the maps of the water flow and pressure distributions***: While operating a waternet it is important for the operator to know the current state of flows and pressures in the network. Using a hydraulic model he can calculate these values but the enormous number of them ordered mostly in the form of big tables makes it quite impossible to recognize quickly in which part of the waternet they are incorrect. Because of that it is useful to give the operator an information tool for a quick assessment of the quality of the waternet work. Such an algorithm is implemented in MACOW. It uses the kriging approximation to design the distributions of water pressures and flows in the form of contour line maps. On these maps the waternet parts with right or bad functioning are marked with different colors (Studzinski and Bogdan, 2007). In Figures 9 and 10 the results of this algorithm are presented. Looking on the maps the waternet operator can recognize very fast the work condition of the network. On our exemplary pictures we can state that the waternet works wrong: the water flows are too slow (these parts are marked with the green and blue colors in Fig. 9) and the pressure values are too high (these parts are marked with the brown and red colors in Fig. 10). In real cases of the waternet management the operator has to undertake in such situations some improvement procedures to decrease the pressure and to accelerate the water flows.

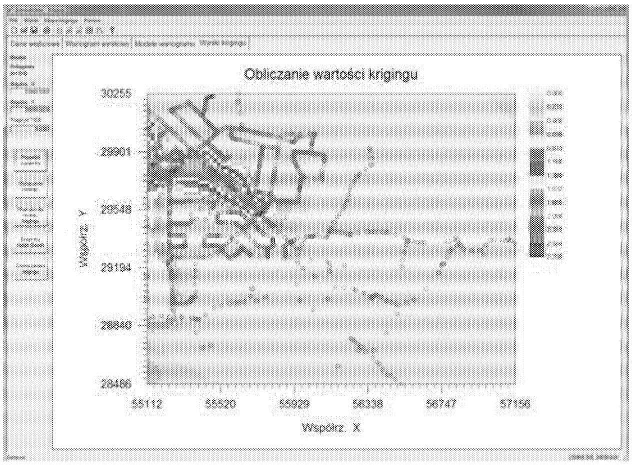


Figure 9: The water flow distribution in the waternet.

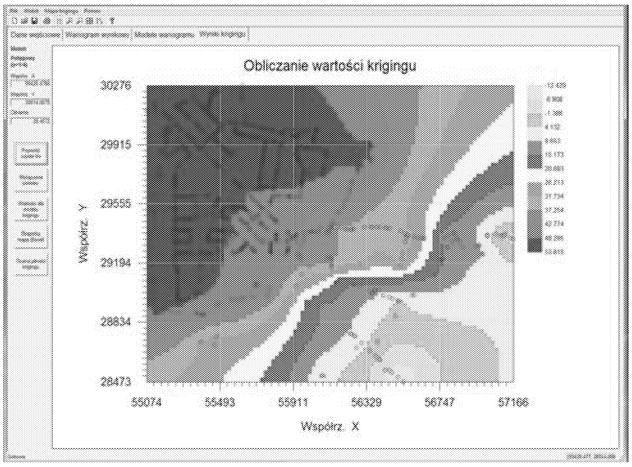


Figure 10: The water pressure distribution in the waternet.

### Conclusions

In the paper the structure of the information system MACOW developed at the IBS PAN to support the management of communal waterworks is presented. This system consists of four modules and their short desriptions have been also outlined. The system works efficiently when different modules cooperate each other and this cooperation is possible while using somme additional tool algorithms. The algorithms presented here enable the intermodular communication, kriging approximation and the calculation of the waternet nodes sensitivity and the use of them expands the scope of tasks that can be solved by MACOW.

The information system presented has been implemented in the Rzeszow waterworks in Poland and the system itself as well as its tool algorithms are still under permanent development. The next algorithms which are now under investigation are dealing with the water leaks detection and with the generation of optimal investment plans for the waternet revitalization.

### REFERENCES

Bogdan L. and J. Studzinski, 2007. "*Modeling of water pressure distribution in water nets using the kriging algorithms*". In: Industrial Simulation Conference ISC'2007 (J. Ottjes and H. Vecke, Eds.) Delft (June) TU Delft Netherlands, 52-56.

Straubel R., 2007. "*REH – Ein Program für Rechnerunterstützte Entscheidungshilfen*". Ingenieurbüro Dr. Straubel, Berlin.

Straubel R. and B. Holznagel, 1998. „*Mehrkriteriale Optimierungen für Planung und Steuerung von Trink- und Abwasser-Verbundsystemen".* In: Proceedings of the Conference on Problems in monitoring and automation of wastewater purification plants. Ustronie Morskie / Poland, 30-42.

Studzinski, J., 2007 "*Computer aided management of waterworks*". In: Proceedings of QRM'2007 (R.A. Thomas, Ed.) 6th Intern. Conference on Quality, Reliability and Maintenance, Oxford.

Bogdan L. and J. Studzinski, 2008. "*Mathematical models for hydraulic calculation and optimization of commnal waternets*". In: Industrial Simulation Conference ISC'2007 (J. Ottjes and H. Vecke, Eds.) Le Havre(June) TU Le Havre France, 52-5

Studzinski J., 2008 "*Rechnergestützte Entscheidungshilfe zur Führung eines kommunalen Wassernetzes*". In: Modellierung und Simulation von Ökosystemen (A. Gnauck, Hrsg.) Shaker Verlag, Aachen (*in print*).

Studzinski J., 2009 "*Waternet modeling and model calibration for the waterworks management*". Reports on the Knowledge Management, Bydgoszcz / Poland (*in print*).

# MODELING AND SIMULATION FOR PREDICTING WATER-FLOWER BASED ON RBF NEURAL NETWORK

Zaiwen LIU,Xiaoyi WANG and Jiping XU
College of Computer and Information Engineering
Beijing Technology and Business University
No.33 Fucheng Road
Beijing 100048
China
Email: liuzw@th.btbu.edu.cn

**KEYWORDS**
Forecasting, Water flower, Modeling, RBF neural network, Environmental science

## ABSTRACT

The research on ecological models of water flower is difficult and complicated, because of the complexity of its mechanism and the effect of human beings. Main factors which make water bloom break out in river and lakes is analyzed, and the modeling method of short-time prediction for water bloom based on RBF neural network, including supervise learning method for the center, width and weight of base function in RBF neural network, error-correction algorithm based on gradient descent of RBF, is proposed. The effect which hidden layer of RBF brings to network performance is compared, and fitting capacity between RBF's width and generalization capability of network is discussed. According to the results of network training and water bloom predict, RBF neural network can be used to predict the change of *Chi_a* （Chlorophyll a） in short term. Because of the strong generalization capability, high predict precision and good fitting performance, the model has established a solid foundation for further research on short-term predict methods of water bloom in river and lakes and the simulation result showed that the method is very practice and useful.

## INTRODUCTION

Water -flower, is caused by the contamination of pools, river, lakes and reservoirs etc., and the increase of eutrophic elements, such as *N, P* etc.. Based on appropriate temperature and illumination, some alga and other hydrophytes increase so immensely that many green or other algal floaters emerge on the surface of water. In water flower, transparence, *DO*(dissolved oxygen) and color of water are changed; water quality is deteriorated; water ageing is accelerated. In all the ecosystem and function of water are damaged badly.

Since water eutrophication emerged, people have been making efforts on predict of water eutrophication and water flower by building ecological models of lakes and rivers. Many eutrophication models with different complicacy have been developed both on theory and practice: from simple model with single state variable, Vollenweider *TP* model to complex ecosystem model with dynamic simulation, for example, Larsen and Welch put forward *P* model of water eutrophication. These models are of great importance on research and management of water eutrophication].

The research on ecological models of water flower is difficult and complicated, because of the complexity of its mechanism and the effect of human beings. At present, most methods are mainly based on the change of influencing factors to predict water flower. However, there are highly nonlinear and uncertain states among all factors of ecosystem, and it is difficult to monitor continuously, therefore, traditional predict methods are seldom valid. Ecological numerical models are considered as the trend of research and predict of water flower and red tide. But the applications of these models are limited, because they depend on the ecological mechanism much.

ANN(Artificial Neural Network) models can be useful to predict water flower; however, there are complexity and uncertainty in water flower, which also make ANN models uncertain. Since water flower emerged in lakes and rivers, many scientific research institutes and universities have researched on eutrophication mechanism of lakes and rivers, and concluded some methods and experiences on controlling water flower. In all, as water qualities are different in different areas, effective methods have still not been found to predict and control water flower accurately. Therefore, it will be better to combine ANN models and mechanism models to predict water flower.

## MODELING FOR WATER FLOWER

### Theory of predicting water flower

A scheme of predicting model for water flower based on soft sensing and ANN. is showed as Fig.1.

### Analyze main factors and select secondary variables

There are mainly three types of indexes of water flower: physical, chemical and biological index. Transparency is the most common physical index. TP and TN, which reflect the potential productivity of hydrophytes, can be considered as the biological indexes of dominant population among biocenosis, in order to help predict and control water flower. In terms of the selection of secondary variables in soft sensing, the influencing factors can be obtained objectively according to ecological condition and test indexes data in different lakes and rivers, as well as general analysis.

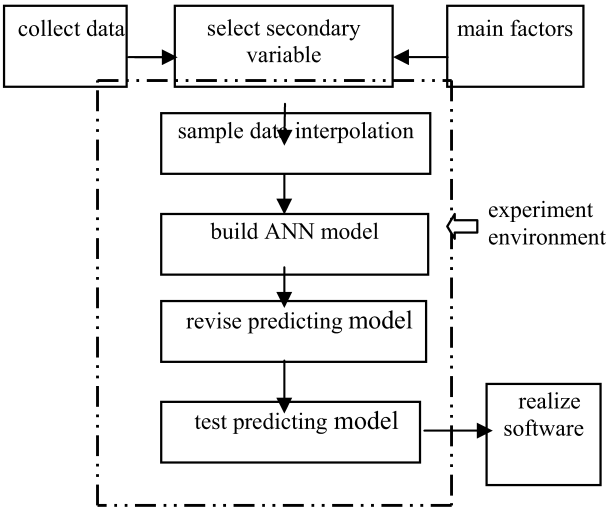


Fig.1 Flow chart of predicting for water flower

In order to ensure the main influencing factors of water flower, compositional analysis can be used to analyze the collected data; the factors with more contribution are taken as the secondary variables of ANN soft sensing method. Data collected should be normalized and analyzed compositionally.

The results are shown in Tab.1.

Tab.1 results of compositional analysis of secondary variables

| | *TW* | *SD* | *DO* | *EC* | *PH* | *TP* | *TN* | *Chi_a* |
|---|---|---|---|---|---|---|---|---|
| 1 | -0.452 | 0.439 | 0.2273 | 0.4395 | -0.335 | -0.127 | 0.1454 | -0.45 |
| 2 | -0.185 | -0.025 | 0.0588 | -0.029 | -0.236 | 0.6671 | -0.674 | -0.06 |
| 3 | -0.008 | 0.1417 | -0.789 | 0.0282 | -0.539 | 0.0468 | 0.1411 | 0.207 |
| 4 | 0.4036 | 0.1324 | 0.1128 | -0.064 | -0.033 | 0.6424 | 0.5694 | -0.25 |
| 5 | -0.177 | -0.498 | -0.203 | 0.7202 | 0.2869 | 0.2335 | 0.1428 | 0.057 |
| 6 | 0.0185 | -0.593 | 0.3987 | -0.045 | -0.655 | -0.099 | 0.1713 | 0.130 |

In Tab.1, the contribution coefficients of *TW, SD, EC, Chi_a* are bigger in 1st composition, and so are *TP, TN* in 2nd composition. Considering these compositions, biological theory of water flower and some data, *TW, SD, EC, Chi_a, TP* are chosen as the input variables of ANN, namely the secondary variables of soft sensing.

Chlorophyll a (*Chi-a*), which generally reflects biomass of phytoplankton, is one of the most important indexes to reflect water flower and its degrees. The biomass and change trend of phytoplankton can be known by analyzing the content and dynamic state of *Chi-a.* Therefore, the content of Chi-a is taken as the output of ANN.

## PREDICTING MODEL OF WATER FLOWER

The first 56 groups of data interpolated in RBF are chosen to be the training data, and the other 4 groups of data are taken as test data. After that, a three-level network with multi inputs and single output can be established, which is shown in the soft sensing model with RBF network as follows.

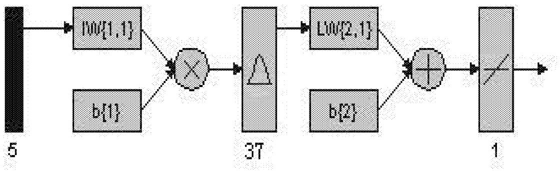


Fig2. predicting model based on RBF

In Fig.2, parameters of soft sensing models are set as follows:

5 Secondary variables: temperature (*TW*), transparency (*SD*), electric conductivity (*EC*), total phosphor (*TP*), chlorophyll (*Chi-a*).

Number of neurons in hidden layer: 37.
1 domain variable in output: *Chi-a.*
Network training precision: 0.001.
Stimulating function in hidden layer: Guess function.
Stimulating function in output layer: linear function.

### Calculation by radial basis function (RBF) neural network

The core of RBF network is the design of hidden layer. The center and width of hidden layer, which represent relative positions of sample space modes and centers, can realize the nonlinear mapping from input layers to hidden layers. The mapping from hidden layers to output layers can be completed by the weight of output layer. The final performance of RBF simply lies on whether the center of radial basis function is selected suitably. The calculation of center of RBF is completed by monitoring learning methods, which are also adopted to train the center, weight and width of RBF. Error correction algorithm based on gradient descent is discussed as follows.
Object function is defined as:

$$E = \frac{1}{2}\sum_{j=1}^{N} e_j^{\ 2} \qquad (1)$$

$$e_j = d_j - F^*(x_j) = d_j - \sum_{i=1}^{m} w_i G(\|x_j - t_i\|_{ci}) \qquad (2)$$

Where *N* is the number of samples, m is the number of hidden units selected, there are three parameters to be learned: $w_{ji}$ , $t_i$ and $\sigma_i^{-1}$ (connected with changing matrix $C_i$ ).

The learning rules of error connection method through gradient descent are shown as follows (n is the number of iterating ).

① the output of weight of unit:

$$\frac{\partial E(n)}{\partial w_i(n)} = \sum_{j=1}^{N} e_j(n) G(\|x_j - t_i(n)\|_{ci}) \qquad (3)$$

$$w_i(n+1) = w_i(n) - \eta_i \frac{\partial E(n)}{\partial w_i(n)} \quad \text{i=1,2,…,m} \qquad (4)$$

② the center of hidden unit:

$$\frac{\partial E(n)}{\partial t_i(n)} = 2w_i(n)\sum_{j=1}^{N} e_j(n) G(\|x_j - t_i(n)\|_{ci}$$

$$\sum\nolimits_i^{-1}(n)\left[x_j - t_i(n)\right] \qquad (5)$$

$$t_i(n+1) = t_i(n) - \eta_2 \frac{\partial E(n)}{\partial t_i(n)} \quad \text{i=1,2,…,m} \qquad (6)$$

③ the width of function:

$$\frac{\partial E(n)}{\partial \sum_i^{-1}(n)} = -w_i(n)\sum_{j=1}^{N} e_j(n) G^{'}(\|x_j - t_i(n)\|_{ci}$$

$$\left[x_j - t_i(n)\right]^T \qquad (7)$$

$$\sum\nolimits_i^{-1}(n+1) = \sum\nolimits_i^{-1}(n) - \eta_3 \frac{\partial E(n)}{\partial \sum_i^{-1}(n)} \quad \text{i=1,2,…,m} \qquad (8)$$

Where $G^{'}(\square)$ is the differential coefficient of Green function $G(\square)$ , $\eta_1, \eta_2, \eta_3$ are learning velocity.

The width of radial basis function is fixed according to the fitting and generalization of network.

**Design RBF and improve its algorithm**

RBF is a forward neural network with two levels, including a hidden layer with radial basis function neuron and an output layer with linear neuron. The advantages of RBF network are simple structure, easy training and fast convergence, through which any nonlinear function can be approached. Therefore, RBF network is widely used in the fields of time sequence analysis and soft sensing with neural network, etc.. Radial basis function is deducted as follows:

If function $h \in L^2(R^d)$ is radial, there will be a function $\Phi \in L^2$ . For $\forall x \in R^d$ , there is a formula

$$h(x) = \Phi(\|x\|) \qquad (9)$$

Where $\|x\|$ is the range of x. According to formula(9), the common expression of radial basis function is:

$$h(x) = \Phi((x-c)^T E^{-1}(x-c)) \qquad (10)$$

Where $\Phi$ represents radial basis function, $c$ represents central vector of function, $E$ is changing matrix.

Radial basis function is a partly distributed, nonnegative nonlinear function to attenuate symmetrically towards center. There are mainly two parameters in this function: one is the center of basis, namely symmetrical point; the other is the width of basis, namely the obvious response to output in most areas. The performance of RBF network mostly depends on the center. RBF with linear parameters can be outspread on the prediction that $\Phi(.)$ and center C are fixed [15]. The common radial basis functions include:

Gaussian Function： $\Phi(t) = e^{-(t^2/\delta^2)}$ (11)

Reflected Sigmoidal Funciton：

$$\Phi(t) = 1/(1+e^{\frac{t^2}{\delta^2}}) \qquad (12)$$

Multiquadric Function：

$$\Phi(t) = 1/(t^2+\delta^2)^a \quad （a>0） \qquad (13)$$

Gaussian Function is in most common use, because of several reasons as follows：

（1）The form of function is simple, even to multi-variable inputs.

（2）Radial symmetry, good smoothness, derivative with any rank exists.

（3）Function is easy to analyze theoretically.

The calculating result of RBF basis function is the Euclid distance between input vector and center; the hidden unit inspiring functions of other networks calculate the internal product between input unit and weight. RBF follows all-purpose approaching principle, which means it can approach continuous functions of compact basis at any precision as long as the data of hidden nodes are enough. The design of hidden node keeps to smallest network structure satisfying the precision, in order to ensure the generalization of network.

However, if there are many input vectors in application, many neurons are needed, which makes network design difficult. Therefore, many scholars put forward some algorithms to improve it. For example: Shenqian etc. presented RBF neural network learning algorithm in which hidden neurons can be “reduced” automatically; Liu meiqin etc. used scaled robust cost function instead of traditional secondary index to search the structure and parameters of optimization RBF neural network (RBFNN), combined with improved genetic algorithm. These improved algorithms have accelerated the application of RBF neural network widely.

## ANALYSIS TO WIDTH OF RBF AND FITTING ABILITY OF NETWORK

The width of radial basis function is a very important parameter in the design of RBF network, which can approach various function to any input and output samples theoretically. However, if these samples are selected improperly, over sufficiency and insufficiency will arise in function approach. Generally speaking, the width is selected according to the distance among input vectors, which is longer than shortest distance and shorter than longest one. If the width constant is too small, more neurons are needed in network, but if the width constant is too large, neurons is same with each other, so that it can not be analyze through network trainings.

Suppose that goal error of network goal=0.001, largest hidden node mn=60, network can be trained through different widths. The fitting ability and generalization performance of network can be observed when width sp is changing, in order to get the best neural network soft sensing model. Fig.3 shows the fitting curve to actual values and predict values, or convergent error curve.

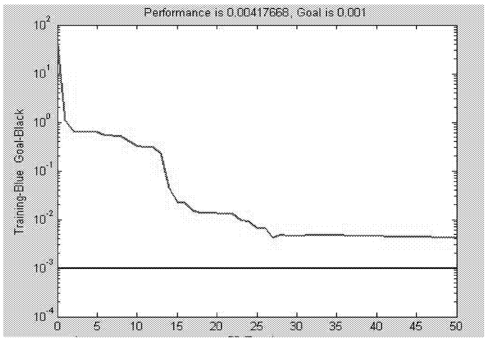


Fig.3: when *sp*=10, error convergence curve (hidden neuron number m=50)

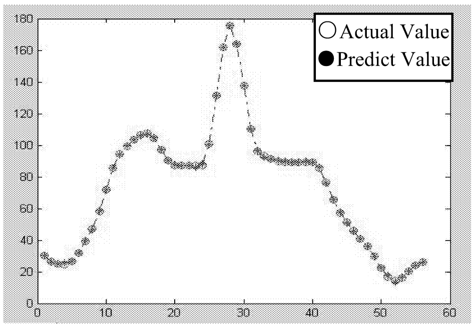


Fig.4: when sp=1, the fitting curve of *chi_a* (hidden neuron number m=36)

From the results of network training, when *sp*=10, network can not converge to expected precision with bad fitting ability.

If reduce the width of radial basis, network can converge to goal precision. Its fitting curves with different widths are shown as follows. In Fig.4-6, y-axis is *Chi_a* (*mg/L*), *x*-axis is samples.

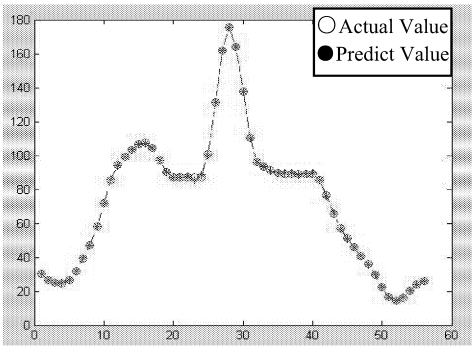


Fig.5: when sp=0.6, the fitting curve of chi_a (hidden neuron number m=38)

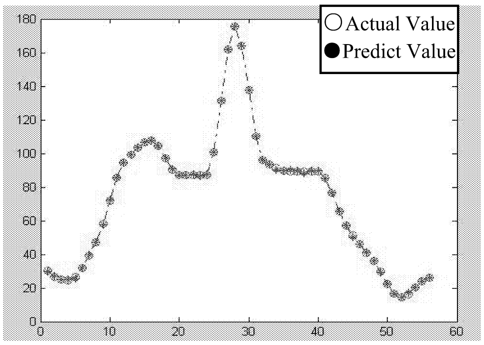


Fig.6: when sp=0.16, the fitting curve of chi_a (hidden neuron number m=37)

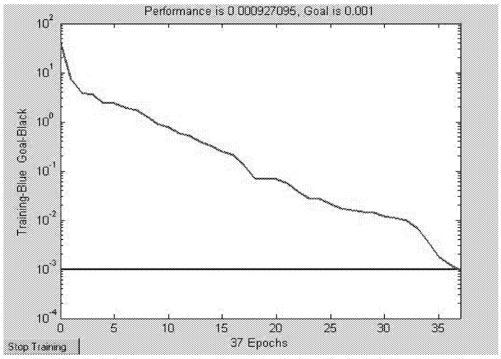


Fig.7: when *sp*=0.16, error convergence curve (hidden neuron number m=37)

From these figures above, it can be concluded that the number of hidden neurons is different along with different widths in RBF trainings. When the width is big, network responds slowly to inputs and outputs. But when the width is small, network has a good fitting performance, in which training data can almost be fit and the trainings are successful.

### Width of basis function of RBF pridictng model and analysis of network generalization ability

There are much difference among generalization abilities of network and predict results of testing data with different widths of basis functions. 4 groups of testing data were used to predict in networks trained with different widths. Results are as follows:

Tab.2 Predict values of *Chi-a* with different widths *Sp*

| Testing data | Predicting value （sp=1） | Predicting value (sp=0.6) | Predicting value (sp=0.16) | Actual measure value |
|---|---|---|---|---|
| Group 1 | 27.44 | 28.82 | 26.80 | 25.9625 |
| Group 2 | 25.01 | 30.88 | 26.83 | 25.7 |
| Group 3 | 17.57 | 28.17 | 25.48 | 24.9875 |
| Group 4 | 9.64 | 26.21 | 23.84 | 23.6 |

Tab.3 Error analysis between predict values and actual values of *Chi_a*

| Testing data | sp=1 | | sp=0.6 | | sp=0.16 | |
|---|---|---|---|---|---|---|
| | Absolute error | Relative error | Absolute error | Relative error | Absolute error | Relative error |
| Group 1 | 1.47 | 5.69％ | 2.85 | 11％ | 0.83 | 3.22％ |
| Group 2 | 0.69 | 2.68％ | 5.18 | 20.15％ | 1.13 | 4.4％ |
| Group 3 | 7.41 | 29.68％ | 3.18 | 12.74％ | 0.49 | 1.97％ |
| Group 4 | 13.96 | 59.15 | 2.61 | 11.1％ | 0.24 | 1％ |

From Tab.2 and 3, it can be seen that: when sp=0.16, the test error of network is smallest; the approaching ability and fitting performance are good; network training is successful. However, in terms of other widths, fitting performances of network are relatively good, but its predict results are bad, which shows that there are “over fitting” and “lack of fitting” in network.

### Test of generalization of RBF soft sensing model

Test aggregate is used to value the performance of trained network. Generally speaking, sample couples included in training aggregate are only a part of data aggregate. Even the network is trained by all the samples in training aggregate, it is still not sure to obtain satisfactory results when the network is tested by other samples. If a group of representative samples, not belong to training samples, is adopted to test network and the result is satisfying, the trained network is of strong generalization ability, otherwise, the training samples chosen are not representative and can not reflect the general characters of data aggregate, so that the network is of weak generalization ability.

Therefore, in order to obtain good network performance, there are two basic preconditions as follows: first, representative samples should be adopted as training and testing aggregates; two, testing aggregate should be different from training aggregate. In this application, the data of measuring points are used to test the generalization of network. In RBF soft sensing model, the width of radial basis function in hidden level sp=0.16, upon which the other parameters of network are obtained by training. Tab.2 shows measuring values of *Chi_a* at measuring points and testing results of predict values in soft sensing model.

Network trained can be used to predict the change of *Chi_a* at measuring points correctly, which shows that the network is of strong generalization and can achieve the expected goal. Fig.8 shows the curves of actual values and predict values, where y-axis is *Chi_a* (mg/L) and x-axis is sample.

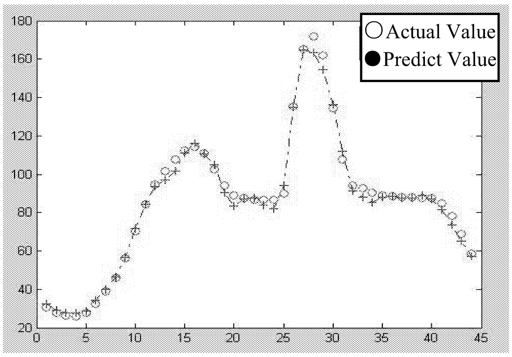


Fig.8 fitting curve of actual values and predict values of *Chi_a* in measuring points

In all, soft sensing model of water flower based on RBF network, which is of strong generalization, precise predict and good fitting performance, provides an effective way to short-time predict and control water flower in lakes and rivers of city. According to the results of net training and predict, RBF network can be used to predict the change of *Chi_a* in short time. This predict method can be effective to predict water flower, play a positive role to deduce the cost of governing water and be directing to protect water resource of cities.

## CONCLUSION

After analyzing and discussing the main factors and evaluating indexes of water flower, we have researched two kinds of short-time predict models of water flower based on BP and RBF neural networks, which are also analyzed and compared. According to the results of research, BP network model tends to become local minimum; training time is long; the speed of convergence is slow; not easy to converge and the approaching ability of function is insufficient. From the results of models, the fitting precision of models is relatively good, but the predict precision is low, and the generalization of network is also not very good to samples which are added newly.

Therefore, short-time predict method of water flower based on RBF network is put forward, including research on the monitoring learning algorithms to the center, width and weight of basis function of RBF network, as well as error-corrected algorithm based on gradient descent. The function and influence, which the number of RBF hidden level nodes brings to the performance of network, are analyzed; the width of RBF and fitting and generalization abilities of network are analyzed and compared. According to the training and predict results, the short-time change of *Chi_a* can be predicted by using RBF neural network; soft sensing model of water flower based on RBF has strong generalization ability, high predict precision and good fitting performance, so that an newly effective method can be provided to predict water flower in short time.

## ACKNOWLEDGEMENTS

This study is supported by the Beijing Natural Science Foundation, and the Beijing Education Committee (PHR201007123、PHR201008238) and its Science and Technology Foundation Project(KM200810011040). Those supports are gratefully acknowledged.

## REFERENCES

Welch E B.Spyridakis D E. Shuster J. Declining lake sediments phosphorus release and oxygen diversion *Journal of Water Pollution Control Fedration* . 1986. 58(1) :92-96.

Somlyody L. Eutrophication modeling, management and decision making: the Kis-Balaton case. *Water Science and Tecnology*, 1998，37(3):165-175.

Vollenweider R A. Input-Output Models with Special Reference to the Phosphorus Loading Concept in Limnology[J].*Schweizeische Zeitschrift Hydrol,* 1975,37:53—84.

Wu H J, Lin Z Y and Guo S L. Application of artificial neural network in predicting resources and environment [J]. *Resource and Environment in the Yangtze Basin,* 2000,9(2):237-241.

Van Gestel T, Suykens J, Viacne S, etc. Benchmarking least squares support vector machine classifiers [J], *Machine Learning*, 2004, 54(1): 5-32.

Dominique M, Alistair B. Nonlinear blind source separation using kernels [J]. *IEEE Trans. on Neural Networks,* 2003, 14(1): 228 -235.

## BIBLIOGRAPHIE

**ZAIWEN LIU** studied control theory and Engineering at the Beijing Institute of Industry, and got PHD. Now he is a professor at the College of Computer and Information Engineering, Beijing Technology and Business University. His main research interests are system modeling and simulation, intelligence control, computer control system and measurement control network.

**XIAOYI WANG** studied control theory and Engineering at the Beijing Institute of Industry, and got PHD. Now he is a associate professor at the Beijing Technology and Business University. His main research interests are system modeling and simulation, intelligence control theory and system.

# SIMULATION TOOLS FOR THE CHOICE OF WATER TREATMENT TECHNOLOGY

Marek M. SOZAŃSKI Andrzej URBANIAK
Institute of Environmental Engineering Institute of Computing Science
Poznań University of Technology
Ul. Piotrowo 2 60-965 POZNAŃ
Poland
E-mail: andrzej.urbaniak@cs.put.poznan.pl

**KEYWORDS**
Environmental modelling, water treatment, simulation tools, data acquisition, real-time simulation.

## ABSTRACT

In the paper there are described the possibilities of complex analysis of fresh water obtained from underground and surface sources, for monitoring and control reasons. The results of the analysis are necessary for preparing the proper technology for drinking water treatment. Full water analysis is a very complex measurement process that can be supported by specialized simulation tools. While constructing measuring process together with technological one it is possible to obtain the results in real time. The proposed simulation software allows to utilize the analysis results for change of technological water treatment processes by choosing the proper structure from the prepared technological scenarios.

## INTRODUCTION

Good quality of drinking water is a very important problem to be solved within a water supply system. The quality of drinking water delivered to consumers depends on fresh water quality, effectiveness of treatment processes and condition of distribution systems (MWH 2005, Sozański et al.1998a). The most important in this chain is a treatment process realized in a water treatment plant. The process depends strictly on pollutions existing in fresh water. Characteristics of underground water are usually stable thus the treatment processes can be established on the basis of laboratory analysis of fresh water. However, after longer exploitation of underground water sources the characteristics of fresh water may change. In case of surface water sources its quality depends more on the present climate change thus there is necessity to change the water treatment technology more often. The traditional approach of designing water treatment technology leads to the search of optimal process technology fulfilling all the requirements concerning drinking water (MWH 2005, Sozański et al. 1998b). However, this approach is non effective in many cases because it is very difficult to change the realized water treatment technology respect to the change of freshwater characteristics. One possible solution of this problem is to create the treatment process as a combination of basic single processes and to choose the proper chain which would finally allow to obtain the good quality of water. The goal will be achieved under two main conditions. First one is to equip a technological installation with controlling devices which help to choose the proper configuration of a treatment process. This requirement is easily being achieved because treatment processes in water treatment plants are very often controlled by using PLC (Programmable Logic Controller). Second condition is to create the possibility of fresh water analysis in time. Basing on this analysis and using data knowledge concerning water treatment technology it is possible to prepare the best configuration of the process of drinking water renewal. Treatment process structure's change consist usually in creation of scheduling for chosen individual (modularly developed) processes. Thus, the whole automation of drinking water preparation demands solving two important problems: automation of data acquisition and algorithms development of configuration change respect to the proper technology. In the paper there is shown the solution for the first problem using professional simulation software for data acquisition and control, taking into account all quality requirements for drinking water.

## SYSTEM STRUCTURE

General structure of the system is presented in the Fig.1. The main tasks of the system are to collect the data concerning parameters of fresh water and its change during water sources exploitation and also to compare the parameters obtained after the treatment processes. The module of data acquisition realises typical algorithms for signal detecting and conditioning, filtration and statistics (Sroczan and Urbaniak 2003, Sroczan and Urbaniak 202). This step can be realized using professional data acquisition software (more detailed information will be given in the next section). The obtained results are utilized for on-line water treatment process evaluation. The final parameters of water delivered to water-pipe network must fulfil all requirements determined by water law concerning drinking water. Structure change of water treatment processes is based on the knowledge about freshwater characteristics (Sozański et al. 1998a, Sozański et al. 1998b)

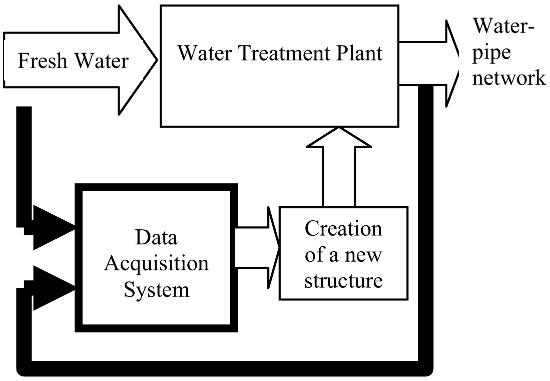


Fig. 1. General structure of the system

The process of searching for an optimal technology solution must consider different quality of two kinds of fresh water: underground and surface water. In case of underground water technological processes are concentrated on iron, manganese and ammonium removal. These tasks are usually realised by aeration and speed filtration processes. In case of surface water the most important task is to remove carbon compounds defined by concentration of total organic carbon (TOC). The task is achieved by processes of coagulation, oxidation and disinfection. As an example, there is presented an application of simulation tools, taking into account the surface water treatment (Sozański et al. 1998a ).

The choice of proper technology of surface water treatment is realized by collecting the four high-effective individual processes in a serial structure depending on results obtained by data acquisition system. The processes are the following:

A. contact coagulation on filter beds,
B. chemical oxidation using ozone and Peroxone,
C. active biological carbon filters,
D. disinfection.

Thus possible configurations of treatment processes can be analyzed in the following systems:

| | |
|---|---|
| System I: | B => A => C => D |
| System II: | A => B => C => D |
| System III: | B => C => D |
| System IV: | A => C => D |
| System V: | C => D |

The choice of the proper system structure is realized by taking into account the optimisation criteria which are minimization of energy consumption during processes and maximization of mass capacity of beds' filtration. Criteria values depend on many technological parameters, such as: method and velocity of filtration, bed's kind and its granulation, high of beds filtration, and others. Choice of parameters in the main process is very difficult and, at the same time, costs' generating. The utilization of specialised simulation software allows to obtain values of process parameters in the proposed parallel systems automatically (Sozański et al. 1998b, Urbaniak 2008).

## SIMULATION TOOLS

The projects of the systems described above were elaborated in two kinds of software: LabVIEW and MATLAB (LabVIEW 2009, MATLAB 2009). Both systems have specialized programming modules for servicing of complex experiment setup. Using hardware and software tools it is possible to realise all functionalities connected with data acquisition and control of the technological processes. The simulation approaches represented by MATLAB and LabVIEW are different but they give possibility to construct the systems fulfilling the requirements characterized above. Simulation systems offer user-friendly graphical interface that allows to utilize them by technologists without very specialized computer knowledge. Producers of simulation systems deliver full documentation connected with effective systems' utilization and special internet service for application creation and servicing ((LabVIEW 2009, MATLAB 2009).

The ordinary MATLAB approach allows to create a control system based on data acquisition obtained from real processes delivered by multifunctional sensors (MATLAB 2009). The algorithms elaborated by technologists make possible the optimal operation of water treatment plant according to technological efficiency as a main criterion. This system can operate as a real-time control system using *Simulink* toolbox. For this reason it is necessary to create the Data Acquisition Session which consists in the following steps:

1. Create a device object

Device objects are the basic toolbox elements that are used to access a hardware device.

2. Add channels or lines

Channels are added to analog input and analog output objects, while lines are added to digital I/O objects.

3. Configure properties

To establish a device object's behaviour.

4. Queue data (analog output only)

Before analog data output, the queue data is performed.

5. Start acquisition or output of the data

Acquisition and output occur in the background, while MATLAB continues executing.

6. Wait for acquisition or output to complete
7. Extract your acquired data (analog input only)
8. Clean up

The overview of Data Acquisition Session presented above is used in many application but it is important to note that fourth step is treated differently for digital I/O objects.

The similar effect can be obtained using LabVIEW approach generating the application diagram (LabVIEW 2009a, 2009b). LabVIEW programs are called virtual instruments (VIs) because their appearance and operation imitate physical instruments, such as oscilloscopes and other devices for measurement of physical magnitudes. Virtual Instrument – VI created in LabVIEW contains three components: front panel, block diagram and an icon with connector pane. The diagram synthesis utilizes G-language which consists of such elements as terminals, nodes, wires and structures.

Terminals represent input and output ports which transport data between front panel and block diagram. The data are introduced using the control elements of the front panel and later are transferred to the block diagram using control terminals.

Nodes are objects on the block diagram that have inputs/ outputs and perform operations.

Wires are utilized for the data transfer between block diagram objects. Wires are different in colours, styles and thicknesses depending on their data types. The wires must be connected with inputs and outputs that are compatible with the data transferred in the wire. The wires must be connected to the only one input and at least one output.

Using LabVIEW we can create test and measurement, data acquisition, instrument control, data logging, measurement analysis and report of generation application.

This is an alternate approach to MATLAB which leads finally to the same effect.

## CONCLUSIONS

The developed approach concerns important adaptation problem of water treatment processes to changeability of the fresh water in the time of operation. There are described the first steps of the problem solution consisting in professional simulation of tools utilization for data acquisition. The carried out analysis let us to formulate some conclusions.

1. The proposed simulation approach allows to choose an effective technology of water treatment respect to requirements formulated for drinking water.
2. Automatically realized measurements in the real time by parallel configured processes give the possibility to change a water treatment process, taking into account present quality of fresh water and its changes.
3. The important result of this solution is a possibility of technological process optimization respect to different criteria, mainly energy consumption.

## REFERENCES

LabVIEW. 2009a. Polish Center LabVIEW, WWW.labview.pl (in Polish)

LabVIEW Graphical Programming Course. 2009b. Collection edited by: National Instruments, Malan Shiralkar, Rice University, Houston, Texas, http://cnx.org/content/col10241/1.4/

MATLAB. 2009. Data Acquisition Toolbox, http:/WWW.mathworks.com

MWH. 2005. Water Treatment Principles and Design (Revised by J.C. Crittenden, R.R. Trussel, D.W. Hand, K.J. Howe and G. Tchobanoglous), John Wiley and Sons, Inc., Hoboken, NJ.

Sozański M.M., Urbaniak A., Rybicki S.A., Jeż-Walkowiak J. 1998a. Conception of pilot station with reseach program for optimization of water treatment technology SUW-Mosina, Poznań. (In Polish)

Sozański M.M., Sroczan E.M., Urbaniak A. 1998b. Tests automation in the experimental-scientific station of water treatment plant, Przegląd Komunalny nr 10 (12), 90-93. (In Polish)

Sroczan E.M., A. Urbaniak, 2003. Intelligent control system for water treatment plant, 12th IASTED International Conference on Applied Simulation and Modelling, - ASM, Marbella (Spain) September 3 - 5, 2003

Sroczan E.M., Urbaniak A., 2002. Intelligent objects' systems in environmental engineering, Technology, Automation and Control of Wastewater and Drinking Water Systems, TiASWiK'2002, IFAC – ASCE, M.A. Brdyś (ed.), Gdańsk, (Poland), 313 – 318.

Urbaniak A.2008. Intelligent systems in water distribution and waste water treatment systems, in: Water Supply and Water Quality, Wyd. PZITS Poznań, ISBN 978-83-89696-20-7. 103-122. (In Polish)

**Andrzej URBANIAK** – born in Poland; studied in the Poznan University of Technology (control engineering) and Poznan Adam Mickiewicz University (mathematics). He obtained the PhD degree in 1979. From 1990 he has been a professor in the Institute of Computing Science. He is an author or co-author of 5 books and over 200 papers concerning the computer control systems and applications of computer science in environmental engineering.

**Marek M. SOZAŃSKI** – studied in the Wroclaw Technical University (environmental engineering), where he obtained the PhD degree in 1975. Today he is a professor in the Institute of Environmental Engineering, (Poznan University of Technology) and supervises the Laboratory of Water Supply and Wastewater Treatment. He is an author of 9 books and over 200 papers concerning water and wastewater treatment technology.

# FLUID
# FLOW
# SIMULATION

# MODELLING AND SIMULATION OF THE CANDU 6 FEEDWATER SYSTEM BEHAVIOUR

Ilie Prisecaru , Daniel Dupleac
Power Plant Engineering Department
Politehnica University of Bucharest
313 Splaiul Independentei, sector 6
RO – 060042 Bucharest, Romania
E-mail: prisec@cne.pub.ro, danieldu@cne.pub.ro

Niță Iulian Pavel
Center of Engineering and
Technology for Nuclear Projects
POB 5204-MG-4, 409 Atomistilor Street
Magurele - Ilfov
E-mail: nitai@router.citon.ro

## KEYWORDS

Nuclear engineering, Model development, Model evaluation, Lumped parameter, Continuous simulation

## ABSTRACT

The paper presents the model developed for the analysis of the feedwater system of the CANDU 6 plant. The model allows the simulation of all the normal and abnormal system transients and can be used to operation procedures validation. The abnormal operation of the system, the case of unavailability of one of the heater, is simulated and analyzed in the paper.

## INTRODUCTION

Demineralised light water is used in the turbine steam and feedwater cycle. Steam, after passing through the turbine, condenses in the turbine condenser and collects in the condenser hotwell, as shown in Figure 1. From there condensate pumps pass the water through a series/parallel network of low pressure (LP) feed heaters to the deaerator. Steam generator feed pumps, pump water from the deaerator storage tank to four steam generators through parallel sets of high pressure heaters and feedwater control valves. At full power the temperature of the feedwater entering the steam generators is 187°C.

The feedwater cycle contains two types of feed-water heaters. There are several horizontal U-tube feedwater heaters and 1 direct contact deaerating feedwater heater. The main source of heat to these heaters is from the turbine extraction steam lines. Feedwater heaters using this source have a self-regulating feature. There are no control valves on the extraction steam supply lines. The steam flow adjusts itself by a thermal equilibrium process. When the feedwater temperature approaches the saturated steam temperature then condensation of the extraction steam diminishes and therefore the flow of extraction steam to the feedwater heater tends towards zero. We can state that generally the steam flow is directly proportional to both the temperature difference and the mass flow of the feedwater.

A secondary source of heat is drain water cascading from high pressure to low pressure heaters and/or moisture separator drains.

Modelling and simulation of the feedwater system behaviour of normal and abnormal operation regimes is a very useful tool for validation of operational procedures. As example, the actual operation procedures stated that if the isolation of one of the LP or HP heater occurs, the rector power must be reduced to 90% of rector full power or a reduction of about 70 MWe of the power generated by the plant.

This study analyzes the case of one heater isolation concurrently with operation of reactor at full power.

The water flow through the system was determined using the calculation code PIPENET [1]. These results are used as input in the Modular Modeling System (MMS) code for the thermal calculations [2].

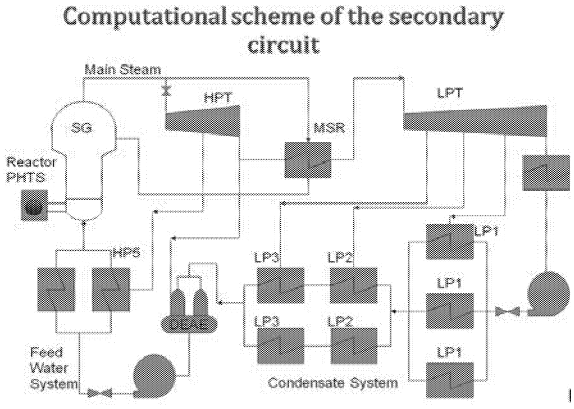


**Fig. 1** CANDU6 feed-water system: SG - Steam Generator; HP - High pressure heater; LP - Low pressure heater; MSR - Moisture separator reaheater; HPT - High pressure turbine; LPT - Low Pressure turbine

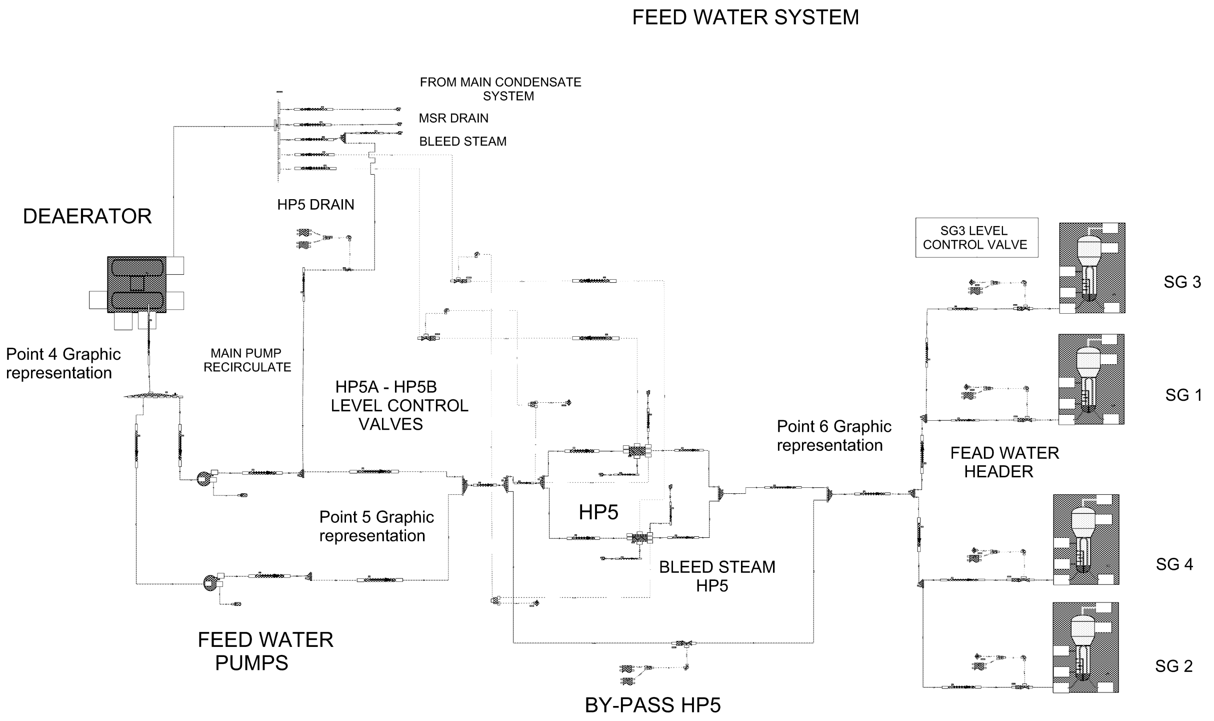


**Fig. 2** The condensate and LP feedwater system model

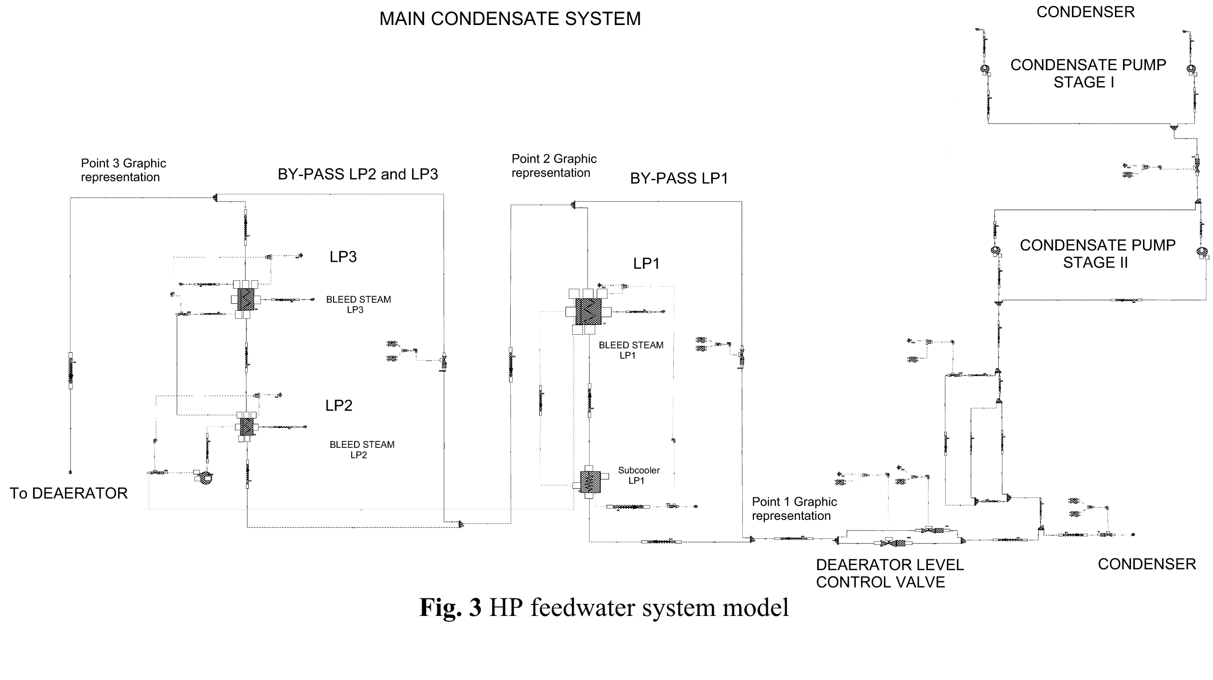


**Fig. 3** HP feedwater system model

## FEEDWATER SYSTEM MODEL

The MMS Flowchart used for the simulation of the CANDU 6 feedwater system is shown in Figures 2 and 3. Detailed mathematical model can be found in references 3 to 7. The details of the heater (U-tube heat exchanger) model are outlined here.

The heater model considers the mass and energy conservation equations. Combining these equations, the general form of the enthalpy variation is obtained of the form:

$$\frac{d\bar{h}}{dt}=\frac{\sum D_I(h_I-\bar{h})-\sum D_E(h_E-\bar{h})+Q+V\frac{dp}{dt}}{\bar{\rho}V+\frac{M_m c_{pm}}{c_{pf}}} \qquad (1)$$

This equation is applied to each characteristic heater region: condensation region and subcooled region [5]. The main assumptions used are:

- saturated steam and condensate are in thermodynamic equilibrium for all regimes;
- the heater shell is considered to have a constant flow area;
- noncondensable gases are neglected;
- pressure losses through shell are negligible.

The model takes into account the following physical phenomena:

- heat transfer between steam, condensate and feedwater;
- condensate level variation inside heater;
- parallel operation of heaters.

Writing the equation (1) for each characteristic region, we obtain the following expressions.

The condensation region

- the tube side

$$\frac{dh_{ae}}{dt} = \frac{D_{ai}(h_i - h_{ae}) + Q_c}{\rho_{ae} V_c + 0.11 M_m} \qquad (2)$$

- the shell side:

$$\frac{dh_h}{dt} = \frac{(D_{si} + D_{di} - D_{de}) - V_h \frac{\partial \rho}{\partial p}\left(\frac{dp_{de}}{dt}\right)}{V_h \left(\frac{\partial \rho}{\partial h}\right)} \qquad (3)$$

and

$$\frac{dp_{de}}{dt} = \frac{\left(\rho_h + h_h \frac{\partial \rho}{\partial h}\right)(D_{si} + D_{di} - D_{de}) + \frac{\partial \rho}{\partial h}(D_{si} h_{si} + D_{di} h_{di} - D_{de} h' - Q_c)}{V_h \left(\frac{\partial \rho}{\partial h} + \frac{\partial \rho}{\partial p} \rho_h\right)} \qquad (4)$$

The subcooled region

- the tube side

$$\frac{dh_i}{dt} = \frac{D_{ai}(h_{ai} - h_i) + Q_s}{\rho_{ai} V_i} \qquad (5)$$

- the shell side

$$\frac{dh_{de}}{dt} = \frac{D_{de}(h_l - h_{de}) - Q_s}{\rho_l V_d} \qquad (6)$$

Consideration of heat transfer across heat exchanger is accounted by the equation:

$$Q_c = \mathrm{k_c} (\mathrm{KA})_\mathrm{c} \Delta T_m \qquad (7)$$

Where

$$\Delta T_m = \frac{(T_s - T_{\mathrm{int}\,rare}) - (T_s - T_{iesire})}{\ln\left(\frac{T_s - T_{\mathrm{int}\,rare}}{T_s - T_{iesire}}\right)} \qquad (8)$$

for the condensation region, and

$$Q_c = \mathrm{k_c} (\mathrm{KA})_\mathrm{c} \Delta T_m \qquad (9)$$

where

$$\Delta T_{ms} = \frac{(T_s - T_i) - (T_{de} - T_{ai})}{\ln\left(\frac{T_s - T}{T_{de} - T_{ai}}\right)} \qquad (10)$$

for the subcooled region.

The physical model and the block diagram of the heater are shown in Figure 4.

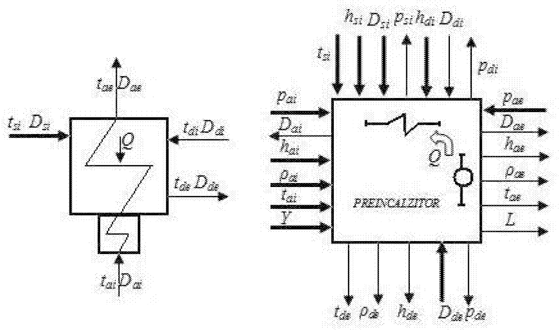


**Fig. 4** Heater model and the block diagram of the model

## RESULTS

Several normal and abnormal operational regimes were simulated. These regimes comprise:

- plant start-up;
- plant shut-down;
- 100 % full power;
- transients from full power to 80% and 60 % of full power;
- reactor shut-down;
- turbine run down.
- by-pass of one of LP or one of HP heater concomitant with reactor power reduction to 90 % full power;
- by-pass of one of LP or one of HP heater maintaining reactor power at 100 % full power

The results obtained for the by-pass of one HP heater are shown in Figures 5 to 10.

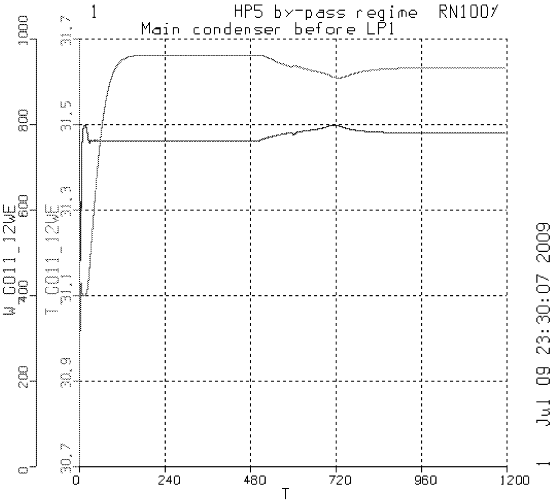


**Fig. 5** Flow rate and temperature upstream of LP1

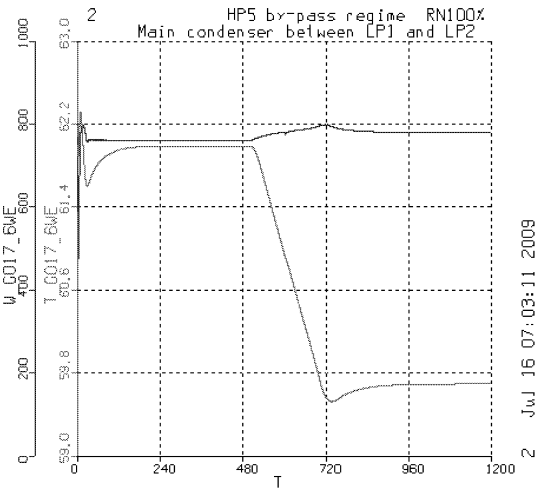


**Fig. 6** Flow rate and temperature on pipe line between LP1 and LP2

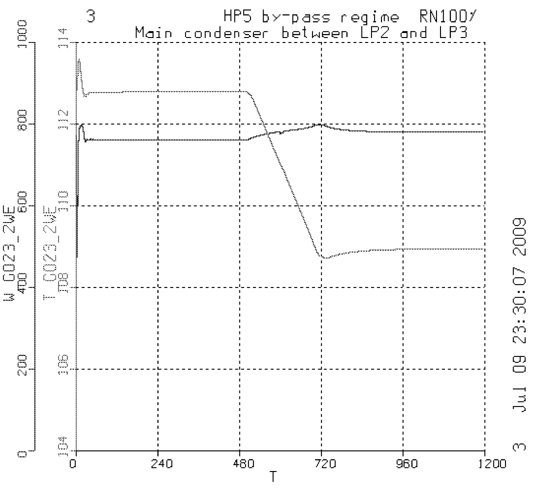


**Fig. 7** Flow rate and temperature on pipe line between LP3 and deaerator

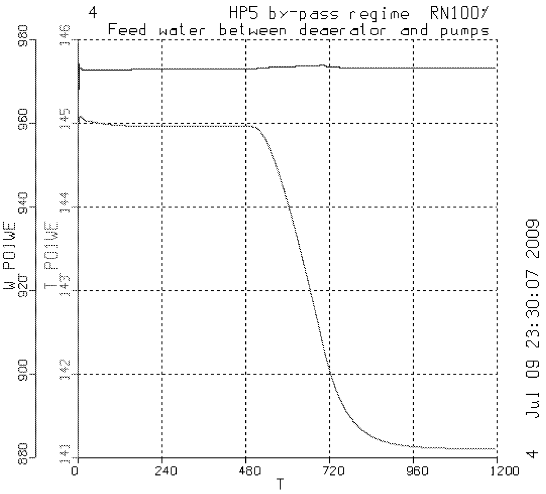


**Fig. 8** Flow rate and temperature on pipe line between deraerator and SG feedwater pump

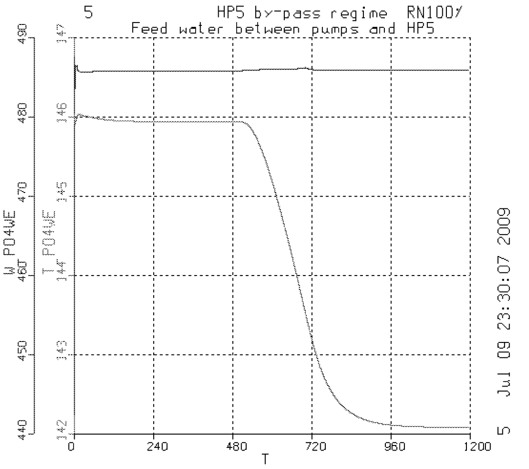


**Fig. 9** Flow rate and temperature on pipe line between SG feedwater pump and HP5

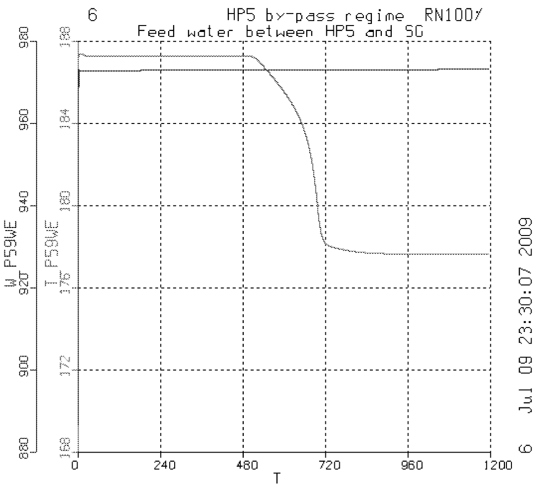


**Fig. 10** Feedwater header flow rate and temperature

The by-pass of the HP5 occurs after 500 s of simulation. After the perturbation, the parameters along the feedwater system become stable after 1000

s. The results obtained show that if the 100% full reactor power is maintained the value of system parameters are within the design limits. In this case, the loss in power generated to the grid is up to 4 % of full power compared to 12% of full power when reactor power is reduced to 90 % full power.

## CONCLUSION

The paper presents the modelling approach of the CANDU 6 feedwater system and the results obtained by simulation of a transient resulting from a HP heater by-pass.

One of the analysis purposes was to validate the normal and abnormal operating procedures of the plant. Also the study of the possibility of improving these procedures was taken into account.

The results obtained show that maintaining the reactor power after the isolation of one LP or HP heater did not affect the safety of the plant. Also this significantly improve the plant performance by generating up to 8 % full power more than in the case in which the reactor power is reduced to 90 % full power as actual operating procedures state.

## REFERENCE

[1] Sunrise Systems Limited. 2000 *PIPENET Users Manual.*

[2] Framatome Tehnologies. 1998a. *MMS Theory Manual Release 5.1.*

[3] Prisecaru, Ilie Modelarea proceselor dinamice din centralele clasice si nucleare , Editura Proxima, Bucuresti, 2009.

[4] Ilie Prisecaru, D. Dupleac, Nita Iulian – " Water Hammer in Nuclear Installations Case Study in Feed WaterSystem" – Proceedings of international conference ESMc'2008

[5] Bigu, M.; Nita, I.; Prisecaru, Ilie; Dupleac, D. Modelling of Preheated Regenerative Chain from NPP Cernavoda using MMS Calculation Code, International Symposium on Nuclear Energy SIEN 2005 p. S1.4.1-S1.4.13, Bucuresti (Romania) 23-27 Oct 2005

[6] Prisecaru, Ilie, Dupleac, Daniel. CANDU Saturated Steam Turbine Modeling using COMPGEN for MMS package. În: vol.: ESMc'2005

[7] Nita, Iulian; Gheorghiu, Mihai; Prisecaru, Ilie; Dupleac, Daniel; Simulating the transient regime for main condensate system at Cernavoda NPP. În: vol.: International Conference on Energy-Environment, CIEM 2005 Bucharest.

## NOTATION

A – heat transfer area, m$^2$;
$c_p$ – specific heat, J/kgK;
$D_{ai}$ - inlet water flow rate, kg/s;
$D_{de}$ - outlet water flow rate, kg/s;
$h_{ae}$ - outlet water enthalpy, J/kg;
$h_{ai}$ - inlet water enthalpy, J/kg;
$h_{de}$ - outlet condensate enthalpy, J/kg;
k – overal heat transfer coefficient, W/m$^2$ $^o$C;
$Q_s$ – heat transferred on subcooling region, W;
$Q_c$ – heat transferred on condenser region, W;
T – temperature, $^o$C;
$V_d$ – shell water volume on subcooling region, m$^3$;
$V_h$ – shell water volume on condenser region, m$^3$;
$V_i$ – shell water volume on condenser region, m$^3$;
$V_c$ – tube water volume on condenser region, m$^3$;
$\rho_{ae}$ – water density at heater outlet, kg/m$^3$;
$\rho_l$ - water density at saturation, kg/m$^3$.

# Parallel State and Noise Estimation of a Nonlinear CSTR Based on a Novel Adaptive Extended Kalman Filter

Lotfollah Jargani
Mehdi Shahbazian
Karim Salahshoor
Vahid Fathabadi
Petroleum University Of Technology
Department of Instrumentation and Automation ,
P.O.Box 12333,Tehran,
Iran
E-mails: pooria_jargani@yahoo.com, shahbazian_m@yahoo.com, Salahshoor@put.ac.ir, vahidfathabadi@gmail.com

**KEYWORDS**
Multi-sensor data fusion, Extended Kalman filter, Centralized Kalman filter, State estimation, Adaptive extended Kalman filter.

## ABSTRACT

The Extended Kalman Filter (EKF) is a prevailing methodology widely utilized for state estimation of nonlinear processes under a noisy environment. The common trend for the EKF implementation assumes pre-specified fixed distribution matrices for both process and measurement noises. This inflexible constant variance set-up which employs the ideal white noise model assumption for describing the process and measurement noises causes the EKF algorithm to diverge or at best converge to a large bound even if the EKF model is perfectly tuned. This paper presents a novel adaptive extended Kalman filter (AEKF) algorithm to cope with the unknown noise variance matrices. Here, however, the variance matrices for both process and measurement noise signals are assumed unknown a priori and thus incrementally estimated and updated using a sliding time window paradigm within which an estimation of the noise variance is calculated and adaptively updated each time the window is shifted forward. The proposed methodology is tested on a simulated continuous stirred tank reactor (CSTR) to estimate 4 states of this nonlinear plant. The simulation results indicate considerable improvements over classical EKF-based and fading extended Kalman filter method.

## INTRODUCTION

Data fusion is a multilevel, multifaceted process dealing with the detection, association, correlation, estimation and combination of data and information from multiple sources to achieve refined state and identity estimation, and complete and timely assessments of situation and threat. The use of multiple sensors allows the data of one sensor to complement that of another sensor in order to extract the greatest amount of information about the sensed environment.(Hall 1992) Among the various techniques available for multi-sensor data fusion, Kalman filtering-based approach is one of the most significant one, as it proves to be an efficient recursive algorithm suitable for real-time applications. Kalman filtering is used in many fields such as control, communication, data assimilation, and target tracking.

The Kalman filter addresses the general problem of trying to estimate the state of a discrete-time controlled process that is governed by a linear stochastic difference equation. But what happens if the process to be estimated and (or) the measurement relationship to the process is non-linear? Some of the most interesting and successful applications of Kalman filtering have been such situations. A Kalman filter that linearizes about the current mean and covariance is referred to as an extended Kalman filter or EKF (Simon 2006).

It is important to note that a fundamental flaw of the EKF is that the distributions (or densities in the continuous case) of the various random variables are no longer normal after undergoing their respective nonlinear transformations. The EKF is simply an ad hoc state estimator that only approximates the optimality of Bayes' rule by linearization. Some interesting work has been done by Julier et al. in developing a variation to the EKF, using methods that preserve the normal distributions throughout the non-linear transformations (Authur et al.1997),(Authur et al. 2004).

All these Kalman filters are used for known noise statistical characteristics. When these statistics are unknown, however, we must use adaptive Kalman filters. We may use adaptive Kalman filter banks for weight to compute variance (Arthur et al. 1987), or output correlation to compute Kalman gain without care for these covariances (Moose 1975), or fading memory algorithm to reduce the effect of prior measurement (Authur et al.2003). These approaches, however, can only be used in linear systems, and reveal poor performances for the case of nonlinear systems.

A novel adaptive Extended Kalman filter (AEKF) is presented in this paper. The main idea of this method is to approximate noise variance by employing a sliding time window within which an estimation of the noise variance is calculated and adaptively updated each time the window is shifted forward.

This paper is organized as follows. In following section , the proposed methodology is presented. Next section describes the CSTR case study. The effectiveness of the proposed approach is demonstrated in next section. Finally, the conclusions are given in last section.

## PROPOSED METHODOLOGY

Multi-sensor data fusion(MSDF) is a synergistic process, concerning the mechanism of fusing uncertain, incomplete, and sometimes conflicting data from a variety of disparate sensors in real time to extract a single compilation of the overall system status for monitoring, control and decision making purposes.

For a particular industrial process application, there might be plenty of associated sensor measurements located at different operational levels and having various accuracy and reliability specifications. One of the key issues in developing a MSDF system is the question of how can the multi-sensor measurements be fused or combined to overcome uncertainty associated with individual data sources and obtain an accurate joint estimate of the system state vector. There exist various approaches to resolve this MSDF problem, of which the KF, for its information form is one of the most significant and applicable candidate solutions (McGee and Schmidt 1985).

### Extended Kalman Filter

Kalman filter use to estimate the state x€R$^{nx}$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_k = F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \quad (1)$$
$$y_k = H_k x_k + v_k \quad (2)$$

The noise processes $\{w_k\}$ and $\{v_k\}$ are white, zero-mean, uncorrelated, and have known covariance matrices $Q_k$ and $R_k$ , respectively:

$$w_k \sim (0, Q_k) \quad (3)$$
$$v_k \sim (0, R_k) \quad (4)$$
$$E[w_k w_k^T] = Q_k\delta_{k-j} \quad (5)$$
$$E[v_k v_k^T] = R_k\delta_{k-j} \quad (6)$$
$$E[v_k v_k^T] = 0 \quad (7)$$

To this point we have considered linear filters for linear systems. However, many practical systems are non-linear. Nonlinear filtering can be a difficult and complex problem. It is certainly not as mature, cohesive, or well understood as linear filtering. There is still a lot of room for advances and improvement in nonlinear estimation techniques. However, some nonlinear estimation methods are becoming widespread. These techniques include nonlinear extensions of the Kalman filter, unscented filtering, and particle filtering. Nonlinear systems can be linearized and then linear estimation techniques (such as the Kalman or H∞ filter) can be applied. This involves finding a linear system whose states represent the deviations from a nominal trajectory of a nonlinear system. We can then use the Kalman filter to estimate the deviations from the nominal trajectory, and hence obtain an estimate of the states of the nonlinear system. The derivation was based on linearizing the nonlinear system around a nominal state trajectory. The question that arises is, how do we know the nominal state trajectory? In some cases it may not be straightforward to find the nominal trajectory. However, since the Kalman filter estimates the state of the system, we can use the Kalman filter estimate as the nominal state trajectory. This is a sort of the bootstrap method. We linearize the nonlinear system around the Kalman filter estimate, and the Kalman filter estimate is based on the linearized system. This idea of the extended Kalman filter (EKF) was originally proposed by Stanley Schmidt so that the Kalman filter could be applied to nonlinear spacecraft navigation problems (Bellantoni and Dodge 1967).

The nonlinear system equations obey the following non-linear relationships:

$$x_k = f_{k-1}(x_{k-1}, u_{k-1}, w_{k-1}) \quad (8)$$
$$y_k = h_k(x_k, v_k) \quad (9)$$
$$w_k \sim (0, Q_k) \quad (10)$$
$$v_k \sim (0, R_k) \quad (11)$$

Where $w_k$ and $v_k$ are process noise and measurement noise with variances of $Q_k$and $R_k$ respectively.

A Taylor series expansion of the state equation will be performed around $x_{k-1} = \hat{x}_{k-1}^+$ and $w_{k-1} = 0$ to obtain the following:

$$\begin{aligned} x_k &= f_{k-1}(\hat{x}_{k-1}^+, u_{k-1}, 0) + \frac{\partial f_{k-1}}{\partial x}|_{\hat{x}_{k-1}^+}(x_{k-1} - \hat{x}_{k-1}^+) + \frac{\partial f_{k-1}}{\partial w}|_{\hat{x}_{k-1}^+} w_{k-1} \\ &= f_{k-1}(\hat{x}_{k-1}^+, u_{k-1}, 0) + F_{k-1}(x_{k-1} - \hat{x}_{k-1}^+) + L_{k-1}w_{k-1} \quad (12) \end{aligned}$$

where

$$F_{k-1} = \frac{\partial f_{k-1}}{\partial x}|_{\hat{x}_{k-1}^+} \quad (13)$$
$$L_{k-1} = \frac{\partial f_{k-1}}{\partial w}|_{\hat{x}_{k-1}^+} \quad (14)$$

Linearization the measurement equation around $x_k = \hat{x}_k^-$ and $v_k = 0$ lead to

$$\begin{aligned} y_k &= h_k(\hat{x}_k^-, 0) + \frac{\partial h_k}{\partial x}|_{\hat{x}_k^-}(x_k - \hat{x}_k^-) + \frac{\partial h_k}{\partial v}|_{\hat{x}_k^-} v_k \\ &= h_k(\hat{x}_k^-, 0) + H_k(x_k - \hat{x}_k^-) + M_k v_k \quad (15) \end{aligned}$$

where

$$H_k = \frac{\partial h_k}{\partial x}|_{\hat{x}_k^-} \quad (16)$$
$$M_k = \frac{\partial h_k}{\partial v}|_{\hat{x}_k^-} \quad (17)$$

A linear state space system and a linear measurement equation are in (12) and (13) respectively. It means that standard Kalman filter equations can be used to estimate the state. Thus, the following equations are named as the EKF equations:

$$\hat{x}_k^- = f_{k-1}(\hat{x}_{k-1}^+, u_{k-1}, 0) \quad (18)$$
$$P_k^- = F_{k-1}P_{k-1}^+F_{k-1}^T + L_{k-1}Q_{k-1}L_{k-1}^T \quad (19)$$
$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + M_k R_k M_k^T)^{-1} \quad (20)$$
$$\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - h_k(\hat{x}_k^-, 0)) \quad (21)$$
$$P_k^+ = (I - K_k H_k)P_k^- \quad (22)$$

It should be noted that both process and measurement noise signals have appeared as additive terms in Eqs (8) and (9) instead of being embedded within the functions $f_{k-1}$ and $h_k$.

**Adaptive Extended Kalman Filter**

The theory presented in previous section makes the Kalman filter an attractive choice for state estimation. But when a Kalman filter is implemented on a real system it may not properly work, even though the theory is correct.

The theory assumes that the system model is precisely known. It is assumed that the noise sequences $\{w_k\}$ and $\{Q\}$ are pure white, zero-mean, and completely uncorrelated. If any of these assumptions are violated, as they always are in real implementations, then the Kalman filter assumptions are violated and the theory may not work. In order to improve filter performance in the face of these realities, the following methods are presented.

*Fading Method*

It is a simple way of forcing the filter to "forget" measurements in the distant past and place more emphasis on recent measurements. This causes the filter to be more responsive to measurements. It theoretically results in the loss of optimality of the Kalman filter, but it may restore convergence and stability. It is better to have a theoretically suboptimal filter that works rather than a theoretically optimal filter that does not work due to modeling errors. The greater responsiveness of the fading-memory filter to recent measurements makes the filter less sensitive to modeling errors, and hence more robust. The main part of fading is updating covariance for each step $k = 1,2,\cdots$ as below:

$$P^{+}_{k(\mathrm{updated})} = \alpha^2 P^{+}_{k} \qquad \alpha > 1 \qquad (23)$$

Note that P is not equal to the covariance of the estimation error. However, the fading-memory filter is more robust to modeling errors than the standard Kalman filter. If $\alpha = 1$ then the fading-memory filter is equivalent to the standard Kalman filter. In most applications, $\alpha$ is only slightly greater than one based on how much past measurements should be forgotten.

*Novel Adaptive Method*

As discussed earlier, KF methods assume that the exact value of measurement and process noise variances are known. However, basically, in real industrial processes, it is not a very practical assumption. Thus, an adaptive method that can estimate the process and measurement noise variances could help to make the assumptions more realistic. Here, we propose to adaptively estimate and subsequently update the noise variance matrices within a sliding time window framework.

Suppose the dynamic system equation is

$$x_k = f_{k-1}(x_{k-1}, u_{k-1}, t_{k-1}) + w_{k-1} \qquad (24)$$

$$\hat{x}^{-}_{k} = f_{k-1}(\hat{x}^{+}_{k-1}, u_{k-1}, t_{k-1}) \qquad (25)$$

Where $\hat{x}^{-}_{k}$ indicates the calculated state using time-update equations and $\hat{x}^{+}_{k}$ shows the estimated state using measurement-update equations. Generally, $\hat{x}^{+}_{k}$ is more accurate than $\hat{x}^{-}_{k}$ due to the fact that the measurement of time K in estimation node is considered in the procedure.

Consider $\hat{x}^{+}_{k}$ is sufficiently close to $x_k$. Hence, following the Eqns. (26) and (27), the process noise can be estimated as the difference between $\hat{x}^{+}_{k}$ and $\hat{x}^{-}_{k}$, i.e. $\widehat{w}_{k-1} = \hat{x}^{+}_{k} - \hat{x}^{-}_{k}$. Consequently, the process noise variance $(\widehat{Q}_{k-1})$ can be obtained via Eq. (28).

$$\widehat{w}_{k-1} = \hat{x}^{+}_{k} - \hat{x}^{-}_{k} \qquad (26)$$

$$\widehat{Q}_{k-1} = E\left[\widehat{w}_{k-1}\widehat{w}^{T}_{k-1}\right] \qquad (27)$$

$Q_k$ was assumed to be diagonal, thus, its estimation $\widehat{Q}_k$ should also be diagonal.

$$\widehat{Q}_{k-1} = \mathrm{diag}[DVQ] \qquad (28)$$

Process noise can be computed using a limited preceding horizon.

$$DVQ_i = \frac{1}{M-1}\sum_{j=1}^{M} \widehat{w}_{i,k-j}\widehat{w}_{i,k-j} \qquad i = 1,2,\cdots,n \qquad (29)$$

M signifies to the length of the window size. Applying Eqns. (28) and (29), the process variance noise can be estimated using previous available data.

The variance of measurement noise should be estimated as well. Generally, the covariance matrix is determined using Eq. (30).

$$P_k = E[(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T] \qquad (30)$$

$\eta_k$ and $\varepsilon_k$ are defined as error in each time-step

$$\eta_k = y_k - H_k\hat{x}^{-}_{k} \qquad (31)$$

$$\varepsilon_k = y_k - H_k\hat{x}^{+}_{k} \qquad (32)$$

In order to estimate the measurement noise, an innovation step is here introduced by authors. In this step, the measurement noise is computed using a new introduced sequence (IS) . IS is defined in Eqs. (33), (34) and (35).

$$IS_{2k-1} = \eta_k \,, IS_{2k} = \varepsilon_k \qquad (33)$$

$$IS = y - H\hat{x} \qquad (34)$$

$$IS = (Hx_k + v_k) - H\hat{x}_k = v_k + H(x_k - \hat{x}_k) \qquad (35)$$

In order to find the relationship between IS and actual measurement noise $R_k$, the variance of IS should be computed. If $v_k$ and $\hat{x}_k$ be independent, $E[IS\ IS^T]$can be calculated using Eq. (36).

$$E[IS\ IS^T] = R_k + HP_kH^T \qquad (36)$$

However, $v_k$ and $\hat{x}_k$ are not really independent, because, as mentioned above, the value of $v_k$ was used in updating $\hat{x}^{+}_{k}$ (See Eq. (21)). Therefore, considering this fact, our suggestion to calculate the variance of IS is presented as follows. The main idea that helped the authors to reach the following formulation was this fact that the variance of the difference between two correlated signals is always smaller than the maximum of signal variances. Extensive simulation tests have shown that these formulas can give us more accurate results.

$$\mathrm{E}[\mathrm{IS}\ \mathrm{IS}^{\mathrm{T}}] = \mathrm{R_k} - \mathrm{H P_k^+ H^T} \qquad (37)$$

$$\widehat{\mathrm{R}}_{\mathrm{k}} = \mathrm{E}[\mathrm{IS}\ \mathrm{IS}^{\mathrm{T}}] + \mathrm{H P_k^+ H^T} \qquad (38)$$

$$\mathrm{E}[\mathrm{IS}\ \mathrm{IS}^{\mathrm{T}}] = \mathrm{diag}[\mathrm{DVR}] \qquad (39)$$

$$DVR_i = \frac{1}{2\mathrm{M}} \sum_{\mathrm{j}=0}^{2\mathrm{M}-1} \mathrm{IS_{i,k-j} IS_{i,k-j}}$$
$$i = 1,2,\cdots,m \qquad (40)$$

## MATHEMATICAL MODEL OF CSTR

An irreversible and exothermic reaction A→B takes place inside the jacket CSTR that is shown in Figure 2 (Sawattanakit and Jaovisidha 1998). The reaction is operated by two proportional controllers that are used to regulate the outlet temperature and the tank level. A cooling jacket surrounds the reactor and the coolant is water in this case. Negligible heat losses, constant densities, perfect mixing inside the tank and uniform temperature in the jacket are assumed.

The dynamic equations describing the system are given by (Jones and Luyben 1989):

$$\frac{dV}{dt} = F_i - F_o \qquad (35)$$

$$\frac{d(VCa)}{dt} = F_i Ca_i - F_o Ca - V\left(k_0 \exp\left(\frac{E_a}{RT}\right)\right) Ca \qquad (36)$$

$$\rho c_p \frac{d(VT)}{dt} = \rho c_p (F_i T_i - F_o T) -$$
$$\Delta H V\left(k_0 \exp\left(\frac{E_a}{RT}\right)\right) Ca - U a_0 (T - T_j) \qquad (37)$$

$$\rho_j V_j c_j \frac{dT_j}{dt} = \rho_j c_j F_j (T_c - T_j) + U a_0 (T - T_j) \qquad (38)$$

$$F_o = 40 - 10(48 - V) \text{ (Level controller)} \qquad (39)$$

$$F_j = 49.9 - 4(600 - T) \text{ (Temperature controller)} \qquad (40)$$

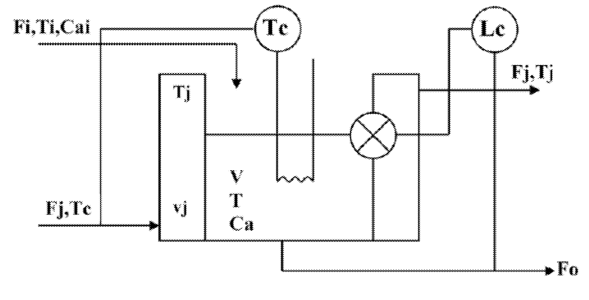


Figure 1. Schematic diagram of the process.

Table 1: Non isothermal CSTR parameter

| Notation | Variable | Steady state values |
|---|---|---|
| $F_o$ | Outlet flow rate | 40 ft$^3$/h |
| $Ca_i$ | Inlet reactant concentration | 0.5 lb. mol of A/ft$^3$ |
| T | Reactor temperature | 600°R |
| $F_j$ | Coolant flow rate | 49.9 ft$^3$/h |
| V | Volume of reactor | 48 ft$^3$ |
| Ca | Reactant concentration in reactor | 0.245 lb.mol of A/ft$^3$ |
| $T_j$ | Jacket temperature | 594.6°R |
| $T_i$ | Inlet feed temperature | 530°R |
| **Notation** | **Variable** | **Parameter values** |
| $V_j$ | Volume of jacket | 3.85 ft$^3$ |
| $E_a$ | Activation energy | 30000Btu/lb.mol |
| U | Heat-transfer coefficient | 150 Btu/h ft$^2$ °R |
| $T_c$ | Inlet feed temperature | 530°R |
| $c_p$ | Heat capacity (process side) | 0.75 Btu/lbm°R |
| ρ | Density of process mixture | 50 lbm/ft$^3$ |
| $k_0$ | Frequency factor | 7.08×1010h$^{-1}$ |
| R | Universal gas constant | 1.99Btu/lb.mol°R |
| $a_0$ | Heat-transfer area | 250 ft$^2$ |
| ΔH | Heat of reaction | -30000 Btu/lb.mol |
| $C_j$ | Heat capacity (coolant side) | 1.0 Btu/lbm°R |
| $\rho_j$ | Density of coolant | 62.3lbm/ft$^3$ |

## SIMULATION STUDIES

For computer simulation, the CSTR nonlinear model is implemented using s-function and SIMIULINK facilities in MATLAB. The basic time unit is hours (hr) and the sampling time is taken to be equal to 0.005 hr.

As it is clear from Fig.2., the outputs of the system are volume and temperature of product, concentration of A, and temperature of CSTR jacket. For the simulation studies, measurements (V, T) have been assumed as the observed values in order to estimate all states of the system (V,T,Ca,Tj).

Proposed methodologies have been implemented on the CSTR plant. Figures 2 to 5 show the estimation of 4 states of the CSTR by 3 methods .The real value of standard deviation of measurement and process noises are 0.002 and 0.005 times of initial real states, respectively. Standard EKF is presented as the comparison criterion in which the correct noise variances are applied. In order to demonstrate the effect of fading and novel adaptive methods in the case of which process and measurement noises are not really known, incorrect values of noises covariance are introduced to these methods to examine their capability to extract the real values. The ratios of the incorrect values to the correct ones are 0.1 and 10, respectively, for R and Q in Figs. 2 to 5. These methods should compensate the effect of shortage in this information.

It should be noted that estimation of concentration state is the most important parameter in output quality of CSTR. However, the simulation of others outputs also verify the capability of the proposed method.

Tables 2-5 can clearly show that the proposed Novel adaptive method is the best choice for industrial cases in which the real values of noise variances are unknown.

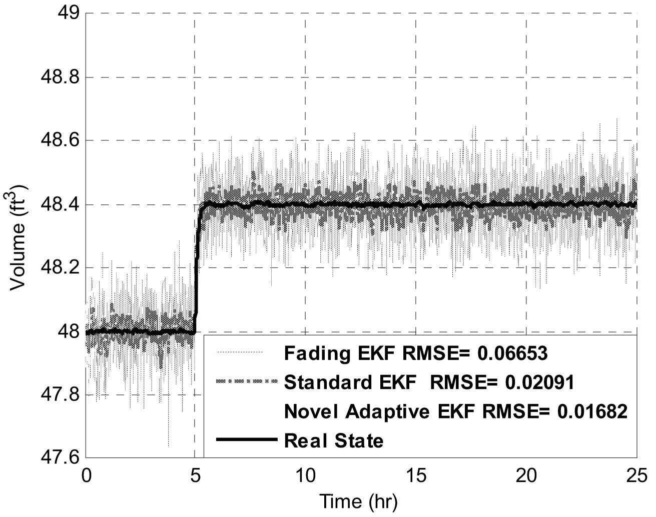


Figure 2.Estimation of volume of CSTR

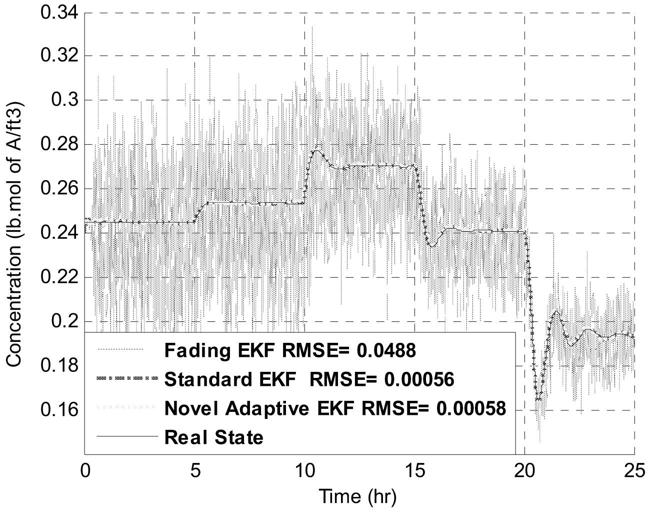


Figure 3.Estimation of product concentration

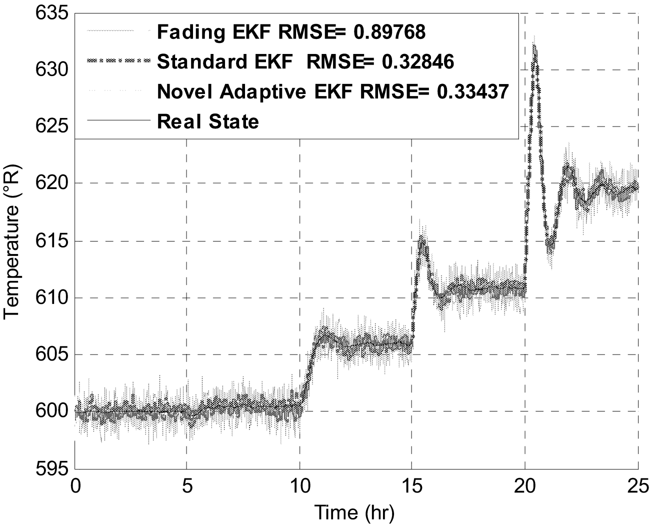


Figure 4.Estimation of Temperature of CSTR

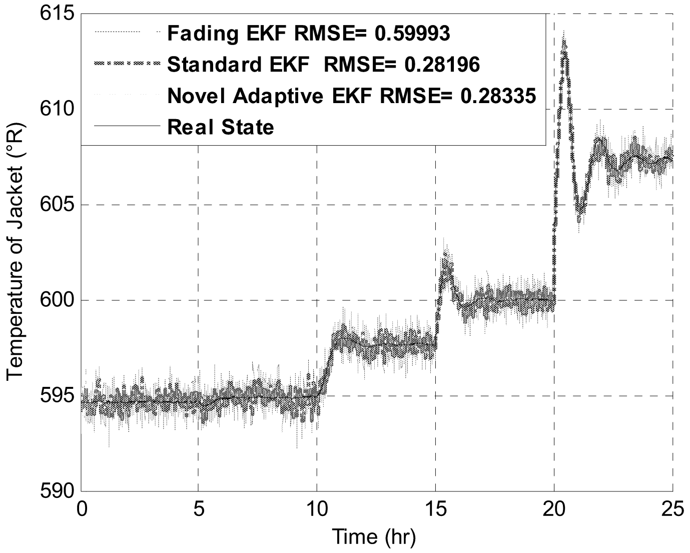


Figure 5.Temperature Estimation of Jacket

## CONCLUSION

The issue of state estimation of nonlinear technical processes under a noisy environment has witnessed an overwhelming research. Highly nonlinear dynamics, unknown or vague knowledge of existing measurement noises, uncertain modeling paradigms, and unavailability of some major state measurements all have been the driving forces to attract many dedicated studies. The Extended Kalman Filter (EKF) is a prevailing methodology widely utilized for this demanding task. The common trend for the EKF implementation assumes pre-specified fixed distribution matrices for both process and measurement noises. This inflexible constant variance set-up which employs the ideal white noise model assumption for describing the process and measurement noises causes the EKF algorithm to diverge or at best converge to a large bound even if the EKF model is perfectly tuned. In this paper a novel adaptive extended Kalman filter (AEKF) algorithm was suggested to cope with the unknown noise variance matrices. Here, variance matrices for both process and measurement noise signals were assumed unknown a priori and thus incrementally estimated and updated using a sliding time window paradigm within which an estimation of the noise variance is calculated and adaptively updated each time the window is shifted forward. The proposed methodology was tested on a simulated continuous stirred tank reactor (CSTR) problem to estimate 4 states of this nonlinear plant. The simulation results indicate notable improvements over the classical EKF-based and fading extended Kalman filter method.

Table 2 : Volume estimation error in terms of RMSE. The first row shows the ratios of incorrect values to correct values of noise variances. The incorrect values are just applied to Fading and Novel Adaptive methods where Standard method uses the actual value of noise and process variances.

| R<br>Method Q | 1<br>0.1 | 0.1<br>1 | 10<br>0.1 | 0.1<br>10 | 0.1<br>0.1 | 10<br>10 | 10<br>0.01 | 0.01<br>10 | 0.1<br>0.01 | 10<br>100 | 100<br>10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Fading | 0.02426 | 0.04085 | 0.03034 | 0.06653 | 0.02936 | 0.03635 | 0.03823 | 0.08849 | 0.03728 | 0.04085 | 0.05125 |
| Standard | 0.02091 | 0.02091 | 0.02091 | 0.02091 | 0.02091 | 0.02091 | 0.02091 | 0.02091 | 0.02091 | 0.02091 | 0.02091 |
| Novel Adaptive | 0.01417 | 0.0158 | 0.01431 | 0.01682 | 0.0148 | 0.01572 | 0.01325 | 0.01749 | 0.01389 | 0.01618 | 0.01583 |

Table 3 : Concentration estimation error in terms of RMSE. The first row shows the ratios of incorrect values to correct values of noise variances. The incorrect values are just applied to Fading and Novel Adaptive methods where Standard method uses the actual value of noise and process variances.

| R<br>Method Q | 1<br>0.1 | 0.1<br>1 | 10<br>0.1 | 0.1<br>10 | 0.1<br>0.1 | 10<br>10 | 10<br>0.01 | 0.01<br>10 | 0.1<br>0.01 | 10<br>100 | 100<br>10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Fading | 0.01412 | 0.02886 | 0.01355 | 0.0488 | 0.01771 | 0.01762 | 0.01345 | 0.06684 | 0.01416 | 0.02867 | 0.01409 |
| Standard | 0.00056 | 0.00056 | 0.00056 | 0.00056 | 0.00056 | 0.00056 | 0.00056 | 0.00056 | 0.00056 | 0.00056 | 0.00056 |
| Novel Adaptive | 0.00062 | 0.00065 | 0.0006 | 0.00058 | 0.00096 | 0.00059 | 0.0006 | 0.00059 | 0.00113 | 0.00058 | 0.00058 |

Table 4 : Temperature estimation error in terms of RMSE. The first row shows the ratios of incorrect values to correct values of noise variances. The incorrect values are just applied to Fading and Novel Adaptive methods where Standard method uses the actual value of noise and process variances.

| R<br>Method Q | 1<br>0.1 | 0.1<br>1 | 10<br>0.1 | 0.1<br>10 | 0.1<br>0.1 | 10<br>10 | 10<br>0.01 | 0.01<br>10 | 0.1<br>0.01 | 10<br>100 | 100<br>10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Fading | 0.5205 | 0.66628 | 0.51461 | 0.89768 | 0.55542 | 0.55461 | 0.51368 | 1.1324 | 0.52081 | 0.66542 | 0.51993 |
| Standard | 0.32846 | 0.32846 | 0.32846 | 0.32846 | 0.32846 | 0.32846 | 0.32846 | 0.32846 | 0.32846 | 0.32846 | 0.32846 |
| Novel Adaptive | 0.32071 | 0.32663 | 0.32052 | 0.33437 | 0.32673 | 0.32246 | 0.32047 | 0.34334 | 0.32784 | 0.32612 | 0.3227 |

Table 5 : Temperature of jacket estimation error in terms of RMSE. The first row shows the ratios of incorrect values to correct values of noise variances. The incorrect values are just applied to Fading and Novel Adaptive methods where Standard method uses the actual value of noise and process variances.

| R<br>Method Q | 1<br>0.1 | 0.1<br>1 | 10<br>0.1 | 0.1<br>10 | 0.1<br>0.1 | 10<br>10 | 10<br>0.01 | 0.01<br>10 | 0.1<br>0.01 | 10<br>100 | 100<br>10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Fading | 0.41554 | 0.49098 | 0.41196 | 0.59993 | 0.43558 | 0.43509 | 0.41139 | 0.72986 | 0.41575 | 0.49063 | 0.41514 |
| Standard | 0.28196 | 0.28196 | 0.28196 | 0.28196 | 0.28196 | 0.28196 | 0.28196 | 0.28196 | 0.28196 | 0.28196 | 0.28196 |
| Novel Adaptive | 0.27474 | 0.2793 | 0.27457 | 0.28335 | 0.27978 | 0.27621 | 0.27452 | 0.28775 | 0.28072 | 0.27885 | 0.27642 |

## REFERENCES

D. L. Hall, *Mathematical technique in multisensory data fusion,* Artech House, Norwood, MA,1992

D. Simon, *Optimal State Estimation Kalman, H∞ and Nonlinear System,* Published by John Wiley & Sons, Inc., Hoboken, New Jersey. Published simultaneously in Canada,2006.

Julier, J. Simon , J. K. Uhlmann "A New Extension of the Kalman Filter to Nonlinear Systems.", *In The Proceedings of AeroSense: The 11th Intemational Symposium on Aerospace Defense Sensing, Simulation and Controls, Mutli Senior Fusion,Tracking and Resource Management II, SPIE*, 1997.

Julier, J. Simon , J. K. Uhlmann "Unscented Filtering and Nonlinear Estimation.",*in Proceedings of the IEEE.* Vol.92, NO.3, March 2004.

R. L. Moose, M. K. Sistanizadeh, G. Skagfjord."Adaptive State Estimation for a System with Unknown Input and Measurement Bias."*IEEE Journal of Oceanic Emgineering*, vol. OE-12, NO. I,Jan. 1987.

R. L. Moose,"An adaptive state estimation solution to the maneuvering target problem."*IEEE Trans, Automat. Contr.*, vol. AC-20, pp.359-362, June 1975.

C. Hu, W. Chen, Y. Chen, D. Liu."Adaptive Kalman Filtering for Vehicle Navigation". *Journal of Global Positioning*, Vol. 2, No. 1: 42-47, 2003.

L. McGee and S. Schmidt, "Discovery of the Kalman filter as a practical tool for aerospace and industry," *NASA Technical Memo 86847,* November,1985.

J. Bellantoni and K. Dodge, "A square root formulation of the Kalman-Schmidt filter," *AIAA Journal*, 5, pp. 1309-1314, 1967

N. Sawattanakit, V. Jaovisidha, "Process Fault Detection and Diagnosis in CSTR System Using Online Approximator".*IEEE*,pp.747-750,1998.

Jones, W. L. Luyben, *Process Modeling Simulation and Control for Chemical engineers,* McGraw-Hill,2nd edition,1989 .

# SIMULATION OF THE 'GP' MTD DEVICE INTENDED FOR THE EXTRACTION OF BLOOD CLOTS BY USING THE BOND GRAPH TECHNIQUE.

Higuera, I. ; Romero, G. ; Félez, J. ; Martínez, M.L.

Graphic Engineering and Simulation Group
ETSI Industriales, Universidad Politécnica de Madrid, C\ José Gutiérrez Abascal, 2, 28006. Madrid, Spain
e-mail {irene.higuera, gregorio.romero, jesus.felez, luisa.mtzmuneta}@upm.es

## ABSTRACT

This article covers the analysis and research into a device recently developed by the University of Wolverhampton (UK), called a *'GP' MTD Mechanical Thrombectomy Device*, under the direction of Dr G. Pearce. This device will improve the process of extracting thrombosis clots in the cerebral arteries. On the one hand, the development of the simulation model of this device is shown by using Bond-Graph formalism and, on the other hand, the optimization of its performance in the very near future, from the interpretation of the results.

## KEYWORDS

Health sciences, Bond Graphs, Blood-clotting, Multi-formalism modelling

## INTRODUCTION

This article Approaches the study and research into a device recently developed under the supervision of Dr G. Pearce from Wolverhampton University (United Kingdom,) called a *'GP' Mechanical Thrombectomy Device (MTD)* (Pearce et al. 2007, Pearce et al. 2008, Pearce et al. 2009, Rai et al. 2009). This device allows improving the process of extraction of typical thrombosis clots in cerebral arteries. Presented in this paper is the development of the model of this device by means of the Bond Graph technique, as well as its simulation and interpretation of the results obtained with the purpose of optimizing its operation for future use.

The aim of the simulation model that is presented is to obtain the minimum pressure necessary to extract the clot and to check that, both this pressure and the time required to complete the operation are reasonable for use in patients, and are in line with experimentally obtained data. It is therefore necessary to consider aspects from the domains of hydraulics and mechanics. The Bond Graph technique (Karnopp et al. 1990) was chosen for its simplicity and suitability for a combined study of both domains.

.

## STATE-OF-THE-ART

Thrombosis is produced by the formation of a clot inside the blood vessels causing an abrupt interruption of the blood flow. In the cerebral arteries this occlusion takes place due to the presence of a clot that has formed at another location of greater diameter which obstructs the cerebral artery due to its smaller cross section.

The process to eliminate this obstruction is called catheterism and different devices exist to carry out this operation. All of them function by introducing a catheter into the artery to eliminate the clot, generally by pushing it. The device under study in this paper, compared to those currently in use, is based on the suction of the obstructing element by creating a vortex, the advantage being that smaller risks are associated with its use. .

For the study of this device, different existing techniques have been considered for the modelling and simulation. Some methods which have been considered for their applicability are the Boltzmann flow simulation technique, and finite elements modelling and its implementation in *Matlab* © software, with 2D or 3D models; or by means of Laplace transformations using Dynamic Motion Solver software.

Finally, the method chosen for the representation and simulation of this model is the Bond Graph technique. Its choice is based on the fact that this technique allows assimilating the model to an electric circuit made up of resistances, capacitances and inductances. Therefore, it is possible to obtain the results in a simple way by evaluating flows and efforts that join and connect the components of the model.

This paper gives a brief description of the device under study, as well as the parts comprising it. Next, the model used for the simulation is described and the phenomena considered to define the device, and, in addition, the values of the parameters used are defined. Lastly the results obtained and the conclusions of this study are attached.

## MODEL DESCRIPTION

The specific device shown in this article is formed by a pump that provides the necessary suction pressure for the operation, joined to a very long catheter. The 'GP' is located at the end of this catheter.

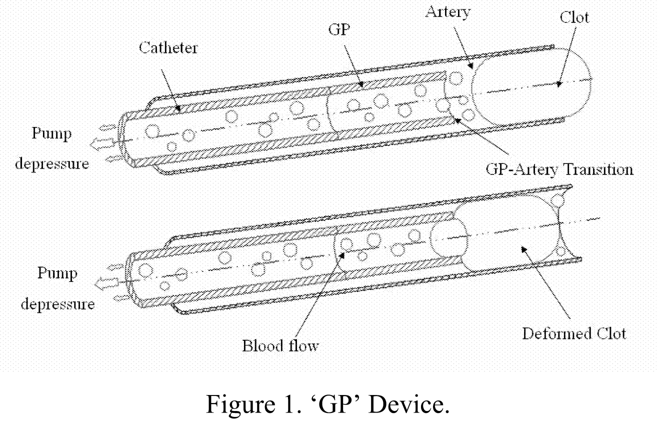


Figure 1. 'GP' Device.

It is a hollow cylinder with the same diameter as the catheter. Its interior is the place where the vortex is created to carry out the extraction. This device is introduced into the cerebral artery at the place where the clot is, and is positioned at a distance of 3mm from it. Then the suction begins until the clot is extracted. The clot crosses the 3mm that separate it from the 'GP' and becomes encrusted with it due to the difference in diameters. Once it is encrusted, the device is removed from the body.

To obtain the simulation of the model, *Bondin* © software will be used (Romero et al. 2009). This program allows obtaining the evolution of the characteristic parameters of the model as well as letting them be compared.

Listed below are the components of the model.

**Pump**

The pump is the component that creates the necessary pressure to carry out the extraction. It is represented by a variable pressure source whose value will increase from zero to 40 kPa, a figure which experience shows to be suitable for carrying out this operation. The time taken to reach 40kPa is 3 sec., after which time the pressure provided by the pump remains constant.

**Catheter and 'GP'**

The catheter is a 110 cm long 1 mm diameter hollow cylindrical tube. It is joined to the 'GP' cylinder of the same diameter and a length of 20mm. In order to represent both elements, they are considered as several pipe sections bearing in mind the different phenomena that take place in their interior: load and inertia loss, and fluid compressibility

Linear load loss is due to the friction between the liquid particles and the pipe walls. Due to their being straight pipes, only linear load losses are taken into account. As this pipe is horizontal and of constant cross section in each section, the load loss is reduced to a pressure loss as the fluid advances along the pipe, the loss being progressive and proportional to the length of the pipe. It is represented by a resistance R, and a type 1 junction.

To determine the equation that governs its behaviour, it is necessary to know if the behaviour of the blood flow is laminar or turbulent. This is evaluated by the Reynolds number, giving the following value:

$$\mathrm{Re} = \frac{V \cdot D}{\nu} \approx 1000 < 2200 \qquad (1)$$

The behaviour is laminar, and can be determined by the following expression:

$$R = \frac{128 \cdot \eta \cdot L}{\pi \cdot D^4} \qquad (2)$$

where η is the dynamic viscosity of the blood flow, L the length of the pipe section and D its diameter.

Secondly, the flow inertia to be overcome in its movement is taken into account. It is represented by a type I, port and a type 1 junction. The expression is:

$$I = \rho \cdot L / A \qquad (3)$$

where $\rho$ is the blood density, L the length of the pipe and A its cross section. Considering this section with circular geometry:

$$A = \pi \cdot \left(\frac{D}{2}\right)^2 \qquad (4)$$

Lastly, the blood compressibility is included. It acts as a spring producing a decrease in volume when the pressure required for compression is increased. This behaviour is dependent on Bulk's blood coefficient (B) and it is defined by a capacitance C with a type 0 junction, by the following expression:

$$K = 4 \cdot B / \pi \cdot D^2 \cdot L \qquad (5)$$

In the model, first the pump is positioned then the catheter. Due to its great length it is represented by ten identical sections that include the three previously described phenomena. Thanks to this representation, it is possible to study the evolution of the pressure loss along the catheter. Later the 'GP' is positioned and is represented by the three previous phenomena.

**'GP'-Artery junction**

The artery is located at the end of the 'GP'. The transition between both elements is considered as a secondary load loss caused by the difference in diameter of both elements and the subsequent variations in flow. These load losses are modelled by a resistance R and can be calculated with the following expression:

$$R = 8 \cdot \rho \cdot \xi \cdot \frac{Q}{\pi^2 \cdot D^4} \qquad (6)$$

where P is the load loss, V the mean speed of the flow in this section and ξ, the load loss coefficient. As the pressures are not equal, the junction is type 1.

The load loss coefficient ξ is an adimensional parameter that quantifies the loss produced and depends on the geometry of the junction. Since this is a narrowing, this value is 0.4.

The flow is not constant during the extraction, so to calculate this expression, its value at each instant is considered like the flow of the inertance that represent the GP device (*flow($I_{GP}$)*).

The diameter indicated in the expression, is the mean diameter between the cylinder and the artery, calculated in the following way:

$$D_{medium} = \frac{D_{cilinder} + D_{artery}}{2} \qquad (7)$$

### Artery

The artery located between the end of the 'GP' and the clot is included in the model as another section of a pipe, similar to the catheter and the 'GP'. It is defined by the loss of linear load (R), the inertia (I) and the compressibility of the blood (C).

In addition, it is necessary to insert a parameter that represents the compressibility of the artery, in line with its Young's modulus:

$$K = \frac{E \cdot h}{V_0 \cdot 2 \cdot r_0} \tag{8}$$

### Domain change

Once all the elements are defined by fluid mechanics, it is necessary to change from the domain of hydraulics to mechanics, to be able to evaluate the movements and efforts in the clot, as well as to define the physical friction between the clot and the artery.

This domain change is carried out by a *Transformer (TF)* element. To calculate the value of the coefficient defining this element, the change in the definition of the flow before and after this element is evaluated. Before the transformer, the flow is in the hydraulics domain, while after, it is in the mechanics domain. The coefficient will be determined by evaluating the required change between both domains. Since the equation that relates both flows is $f_2 = f_1 \cdot r$, where $f_1 = Q_1 = v_1 \cdot A_1$ and $f_2 = v_2$, then,:

$$r = {}^{1}\!/_{A_1} = {}^{1}\!/_{\pi \cdot R^2} \tag{9}$$

where R is the artery radius.

### Clot

Representing the clot is the most complex part of the model. Firstly, the existence of a spring is considered (C port). It measures the force support by the beginning of the clot. Experimental data indicate that the clot begins its movement when this force is equal to 0.01N, from which the value of the constant of this spring is determined by the expression:

$$K = F / q \tag{10}$$

Additionally a resistance R is positioned. It represents the friction between the clot and the arterial wall. The value of this parameter is variable depending on whether the clot has not begun its movement (static friction) or if it is already in movement (dynamic friction). This value is obtained starting from the Stokes equation.

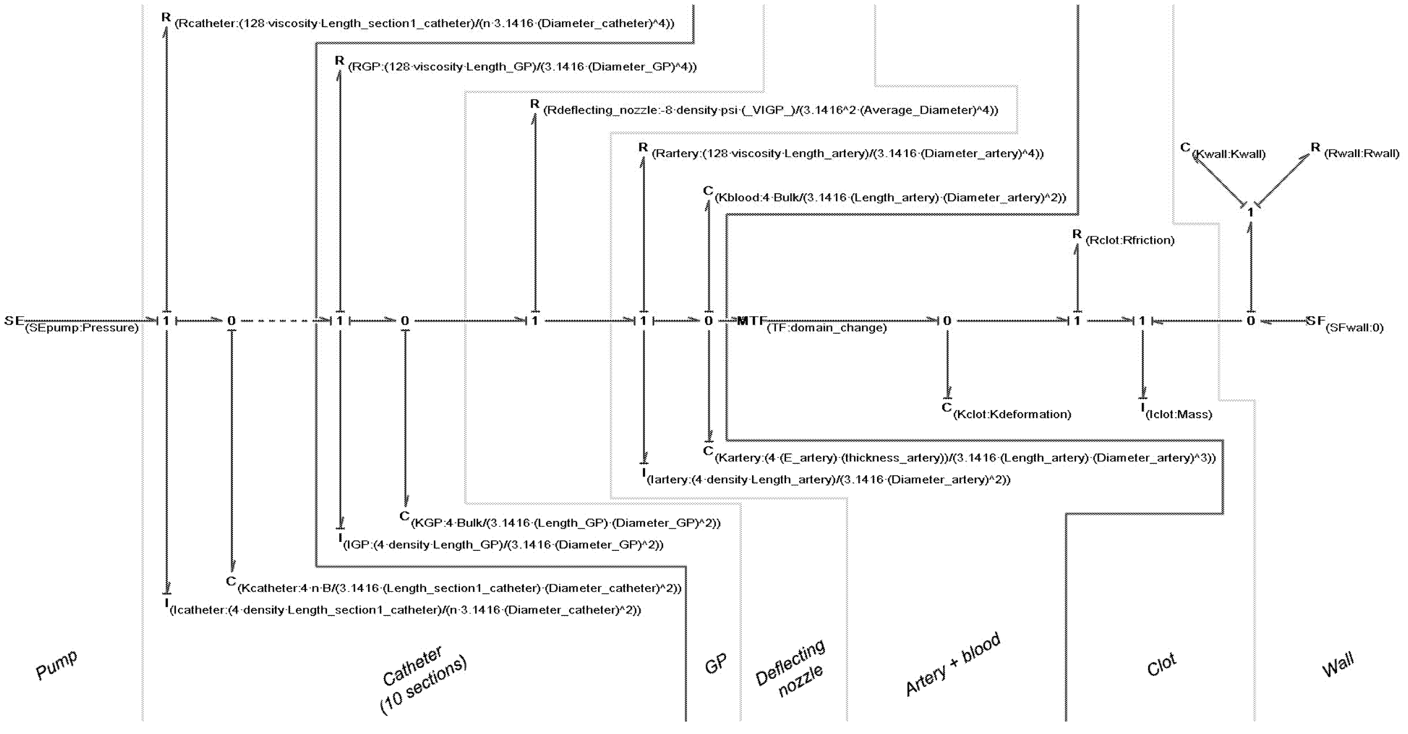
An inertia is inserted that represents the mass of the clot.

$$I = m \tag{11}$$

Finally, a spring-damper system is used to ensure that the clot remains at rest while the force existing at its beginning is less than 0.01 N. The spring-damper system is joined to a zero flow source. While the clot does not receive the force of minimum suction, it has a zero speed. However, when it begins its movement, the spring-damper system is cancelled allowing its extraction.

### Complete model representation

The implementation of the model required for the simulation is shown and it has been made connecting the different components show previously. The full model is presented divided into three ordered blocks.



R (Rcatheter:(128·viscosity·Length_section1_catheter)/(n·3.1416·(Diameter_catheter)^4))
R (RGP:(128·viscosity·Length_GP)/(3.1416·(Diameter_GP)^4))
R (Rdeflecting_nozzle:-8·density·psi·(_VIGP_)/(3.1416^2·(Average_Diameter)^4))
R (Rartery:(128·viscosity·Length_artery)/(3.1416·(Diameter_artery)^4))
C (Kblood:4·Bulk/(3.1416·(Length_artery)·(Diameter_artery)^2))
C (Kwall:Kwall)
R (Rwall:Rwall)
R (Rclot:Rfriction)
SE (SEpump:Pressure)
1
0
1
0
1
1
0
MTF (TF:domain_change)
0
1
1
0
SF (SFwall:0)
1
C (Kclot:Kdeformation)
I (Iclot:Mass)
C (Kartery:(4·(E_artery)·(thickness_artery))/(3.1416·(Length_artery)·(Diameter_artery)^3))
I (Iartery:(4·density·Length_artery)/(3.1416·(Diameter_artery)^2))
C (KGP:4·Bulk/(3.1416·(Length_GP)·(Diameter_GP)^2))
I (IGP:(4·density·Length_GP)/(3.1416·(Diameter_GP)^2))
C (Kcatheter:4·n·B/(3.1416·(Length_section1_catheter)·(Diameter_catheter)^2))
I (Icatheter:(4·density·Length_section1_catheter)/(n·3.1416·(Diameter_catheter)^2))
Pump
Catheter (10 sections)
GP
Deflecting nozzle
Artery + blood
Clot
Wall

Figure 2. Complete model.

First block shows the junction between the pump with the catheter. Since this is a repeated a section, only first section are shown, each one characterized by the previously described R, C and I ports . Second block includes the 'GP', the loss between the 'GP' and the artery, and the artery and blood. Finally, the third block represents the parameters describing the clot and its movement, which includes the change from hydraulics to mechanics.

## MODEL VALIDATION

The object of this study consists in determining the minimum pressure required for the extraction of a blood clot. To do this, by varying the values of the pressure source, the movement of the clot and the time required for its extraction are measured, thereby obtaining the optimum minimum pressure.

To carry out the model validation, the values of the parameters used in the simulation are listed in the following table.

| | |
|---|---|
| Pressure | 0 - [-40, -60] kPa |
| Blood Viscosity ($\eta$) | 0.0035 Pa·s |
| Blood Density ( $\rho$ ) | 1060 kg/m³ |
| Bulk's coefficient | 2200000000 N/m |
| Catheter length (L) | 110 cm |
| Catheter diameter (D) | 0.001 m |
| 'GP' length (L) | 0.020 m |
| 'GP' diameter (D) | 0.001 m |
| 'GP' thickness (h) | 0.0001 m |
| Artery Young modulus (E) | 2800000000N/m |
| Artery thickness (h) | 0.0001 m |
| Artery diameter (Da) | 0.003 m |
| Artery length (La) | 0.003 m |
| Load loss coefficient ($\xi$) | 0.4 |
| Flow load loss (Q) | flow($I_{GP}$) |
| 'GP'-artery mean diameter (Dm) | 0.002 m |
| Domain change coefficient (r) | 141471.06000 |
| Static friction | 0.0000025 N·s/m |
| Dynamic friction | 0.000000025 N·s/m |
| Clot weight | 0.001 kg |

Table I. Parameter values

Likewise, by introducing gauges into the model, pressure loss is evaluated through the sections determining where these losses are concentrated. Also evaluated is if the artery possesses the necessary strength to support the pressure to which it is subjected in this operation.

## RESULTS

The results obtained are shown after the model simulation using a 40 kPa suction pressure and considering a 5cm long obstructive clot of 1gr. mass.

In figure 3 the evolution of the suction pressure supported by the clot can be observed. It can be seen that it undergoes an increase over time until it reaches a value of 1.41 kPa at 112 seconds. This pressure corresponds to 0.01 N, the point at which the clot starts its movement.

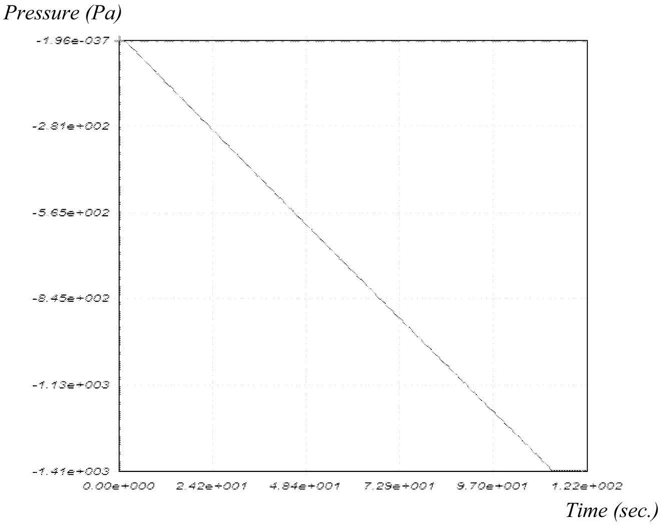


Figure 3. Clot pressure at beginning of movement.

Figure 4 shows clot movement. It can be seen that its movement is zero until 112 seconds, a point at which it reaches a force of 0.01N. From this instant it begins its movement. The clot must travel 3mm through the artery, which is the distance that separates it from the end of the 'GP', in which the clot will be encrusted. It can be observed that the clot needs 114 seconds. to travel this distance.

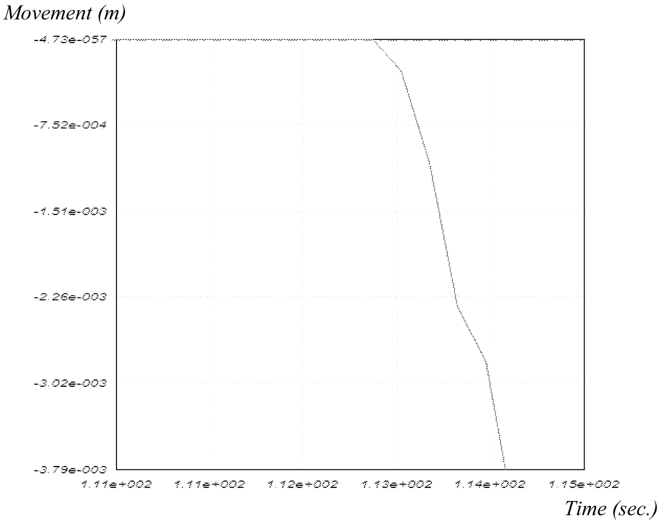


Figure 4. Clot movement.

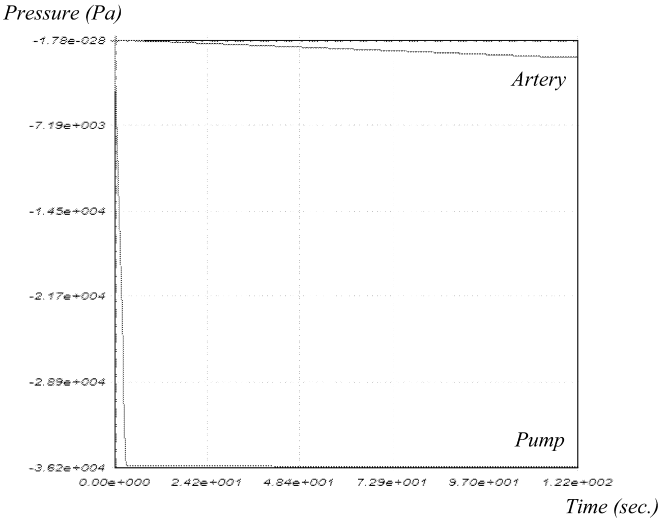


Figure 5. Pressure loss between the pump and the artery.

Figure 5 shows the pressure loss between the pump outlet and the artery.

A significant pressure loss occurs as can be seen in figure 6. It is mainly concentrated in the catheter joining the pump to the 'GP'. As the blood is sucked away, the device progressively loses pressure in each section it flows through due to load loss, blood compressibility and to the blood's own inertia.

Figure 6 shows the loss in pressure in three catheter sections, with a behaviour proportional to the distance being observed.

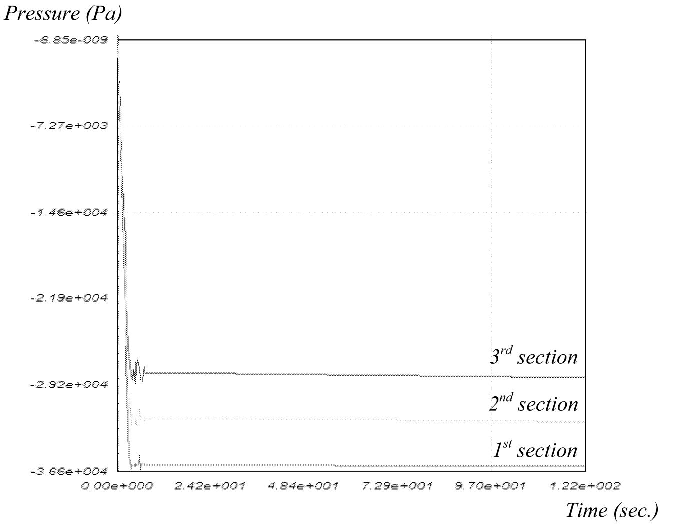


Figure 6. Loss in pressure in the first sections of the catheter.

Finally, in figure 7 the loss in pressure is shown from the pump up to the clot . The proportionality of the loss in pressure is observed in the catheter and its greater magnitude in respect of the loss experienced in the 'GP' and artery.

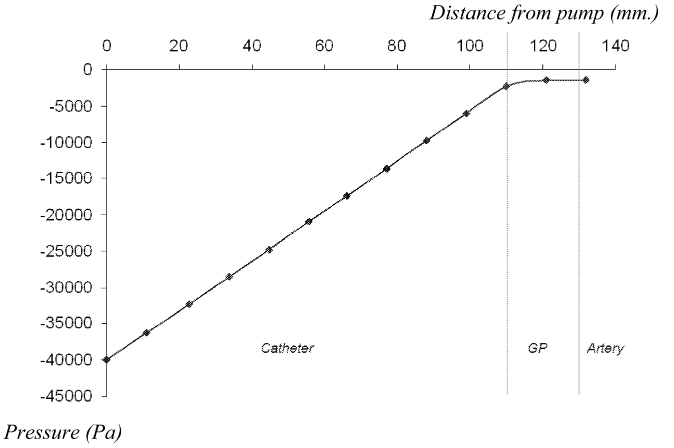


Figure 7. Loss in pressure in the catheter, GP and artery.

Previous figure has been obtained when the clot is moving and it´s possible see that the loss of pressure appear mainly in the catheter part due to the length of this component and how the loss of pressure is the same in each division too. After this loss of pressure, the pressure are 99% stabilized in the GP (the less of pressure are lower than in the catheter). Finally a little percentage of the de-pressure pump remove the clot.

## CONCLUSIONS

Experimental data indicate that the extraction of the blood clot takes place in an interval of time of [60,120] sec. In figure 4 it can be seen that the simulation produces the extraction in 114 seconds, a figure that is coherent with experience.

The model indicates that a pressure of 40 kPa is enough for the extraction of a 5cm long 1g clot. It would be necessary to check how this pressure evolves by varying the mass and length of the clot.

On the other hand, studies demonstrate that the artery has a resistance of 750 mmHg, which is equivalent to 100 Kpa. In figure 3 it is seen that the pressure in this area rises to 1.41 kPa, there being a wide danger margin for rupture of the artery.

Finally it is shown that an important pressure loss takes place in the catheter joining the pump to the 'GP'. These values obtained can be used to optimize its geometry.

## FUTURE WORKS

This work is a first attempt to optimize the operation of the 'GP' device, ensuring its future applicability and compatibility with the experimental data obtained by Dr Pearce. The subsequent lines of work should focus on developing a highly accurate model, the study of different clot sizes and the rechanneling of the blood flow after clot removal.

## REFERENCES

Karnopp, D.C., Margolis, D.L. and Rosemberg, R.C. 1990. "System Dynamics: A Unified Approach". John Wiley & Sons, Inc., Second edition.

Pearce, G., Patrick, J.H. and Perkinson, N.D. 2007. "A New Device For the Treatment of Thromboembolic Strokes", Journal of Stroke and Cerebrovascular Diseases*,* Vol. 16, No. 4, pp 167-172.

Pearce, G., Alyas, S., Perkinson, N.D. and Patrick, J.H. 2008. "Modelling of the 'GP' Mechanical Thrombectomy Device MTD", 10th International Conference on Computer Modeling and Simulation, pp. 499-502. Cambridge, UK.

Pearce, G., Jaegle, F., Gwatkin, L., Wong, J., Perkinson, N.D. and Spence, J. 2009. "An Investigation of fluid flow through a modified design for the 'GP' device", 11th International Conference on Computer Modelling and Simulation, pp.191-195. Cambridge, UK.

Rai, M., Pearce, G., Perkinson, N.D., Brookfield, P., Asquith, J., Jadun C., Wong, J. and Burley, M. 2009. "A Versatile Low cost Arterial Simulator", 11th International Conference on Computer Modelling and Simulation, pp. 196-199. Cambridge, UK.

Romero, G., Felez, J., Cabanellas, J.M. and Maroto, J. 2009. "BONDIN: a new engineering simulation software for ODE and DAE systems with symbolic notation based in the bond graph technique", 8th WSEAS Int. Conf. on Software engineering, parallel and distributed systems, pp. 90-97. Cambridge, UK.

## BIOGRAPHY

**IRENE HIGUERA** received her Mechanical Engineering degree from the Technical University of Madrid (Spain) in 2009. She has worked as researcher at U.P.M. since 2008. She developed her research in the field of simulation techniques of multi-domain systems based on bond graph methodology. She has been involved in others tech papers.

**GREGORIO ROMERO** received his Mechanical Engineering degree from the UNED (Spain) in 2000. He got his PhD Degree from the Technical University of Madrid in Spain in 2005 working on simulation and virtual reality, optimizing equations systems. He started as Assistant Professor at the Technical University of Madrid in Spain (UPM) in 2001 and became Associated Professor in 2008. He is developing his research in the field of simulation and virtual reality including simulation techniques based on bond graph methodology and virtual reality techniques to simulation in real time. He has published more than 40 technical papers and has been actively involved in over 25 research and development projects and different educational projects.

**JESÚS FÉLEZ** received his Mechanical Engineering and Doctoral degrees from the University of Zaragoza in 1985 and 1989. He started as Associate Professor at the Technical University of Madrid in Spain (UPM) in 1990 and became Full Professor in 1997. His main activities and research interests are mainly focused on the field of simulation, computer graphics and virtual reality. His research includes simulation techniques based on bond graph methodology and virtual reality techniques, mainly addressed towards the development of simulators. He has published over 60 technical papers and has been actively involved in over 35 research and development projects. He has served as thesis advisor for 30 master's theses and four doctoral dissertations.

**M. LUISA MARTÍNEZ** received her Mechanical Engineering and Doctoral degrees from the Technical University of Madrid (Spain) in 1990. She got her PhD Degree in 1997 working on variational geometry. In 1990 she started to work as Associate Professor at the Technical University of Madrid in Spain (UPM). Her thesis was focused on variational geometry. She usually works in the field of computer graphics, virtual reality and CAD. During this time she has been involved in different educational projects and pilot activities promoted by the European Commission and other Spanish institutions. She has published over 45 technical papers and has been actively involved in over 30 research and development projects.

# PLUME SIMULATION

# BAYESIAN TRACKING OF THE TOXIC PLUME SPREADING IN THE EARLY STAGE OF RADIATION ACCIDENT

Petr Pecha, Radek Hofman and Václav Šmídl
Institute of Information Theory and Automation of ASCR
Pod Vodarenskou vezi 4
182 08 Prague 8, Czech Republic
E-mail: pecha@utia.cas.cz

**KEYWORDS**
Pollutant Spreading, Data Assimilation, State-Space models, Particle Filtering, Resampling

## ABSTRACT

The article deals with the predictions of time and space evolution of pollution dispersion during the early phase of a hypotetical radiation accident. The goal is to design a proper fast algorithm which could enable more precise online estimation of radioactivity propagation on basis of recursive procedure of Bayesian filtering. Predicted trajectory of the plume of pollutants is refined online according to the values of observations incoming from terrain. The technique should be sufficiently robust to cope an expected lack of information in the same beginning of the event. A certain modification of the particle filter (PF) method is investigated here. Its robustness is illustrated on a real but atypical meteorological situation. Short time meteorological forecast entering the model is for this case in poor correspondence with the real time local meteorological measurements. Radiological measurements are assumed to be coming periodically from the Czech Early Warning Network (EWN). The respective radiological values in the real positions of EWN receptors are generated "artificially" drawing inspiration from the real local meteorological measurements.

## INTRODUCTION

Ongoing efforts on improvement of safety requirements cover both implementation of inherent safety features of the new constructed facilities and substantial improvement of emergency preparedness and response. Tracking and predictions of hazardous material spreading through the living environment provide decision-makers fundamental information for effective emergency management. Modelers should be capable to generate relevant information even in the lack of some basic input information. Correct chain of simulated consequences requires as realistic as possible description of the accident evolution from the same beginning of the harmful substances release. Just at the moment the accident scenario is not known completely and large uncertainties are involved. The evolution of emergency situation is usually so far varied and complicated that specific ad hoc solutions have to be introduced.

In this paper we are studying an application of data assimilation (DA) procedure insisting in optimum combination of prior knowledge with real observations incoming from terrain. The observations bring simultaneously an indirect information related to the system state. Advanced statistical assimilation methods account for both model and measurements error covariance structure. The problem of pollution spreading in the atmosphere is described by nonlinear and generally non-Gaussian model. The attention is focused on Bayesian tracking of the toxic plume propagation over the terrain. It was shown (e.g. Doucet et al. 2001, Doucet et al. 2008, Hoteit et al. 2008, Moradkhani 2008) that except simple problems the Bayesian inference in such complex systems is not analytically tractable.

Consequently, the technique implemented here tries to solve a certain particular task of recursive Bayesian filter by Monte Carlo simulations. The objective of tracking is to refine recursively model predictions on basis of incoming measurements. Tracking in Bayesian approach concerns of recursive evaluation of the state posterior probability density function (*pdf*) evolution based on all available information. The article addresses the Bayesian tracking procedure from the same beginning of the complicated toxic plume spreading under (possibly) incomplete scenario description.

## PROBLEM FORMULATION

We restrict our attention to the stochastic state-space models

$$\begin{aligned} x_t &= b(x_{t-1}) + w_t \\ y_t &= h(x_t) + v_t \end{aligned} \qquad (1)$$

in discrete time steps $t=1,...,T$. Here, $\boldsymbol{x}_t$ is N-dimensional vector unobserved internal quantities describing state of the model at time $t$, and $\boldsymbol{y}_t$ is M-dimensional vector of measurements obtained during the time step $< t\text{-}1;t >$. Nonlinear vector functions $b()$ and $h()$ describe evolution of the state in time, and mapping of the state to measurements, respectively. Disturbance (noise) vectors $w_t$ and $v_t$ are considered to be independent realizations of random variables with zero mean and known variances, $Q_t$ and $R_t$, respectively.

Formalization (1) is intuitively appealing for stationary additive disturbances (noises). However, it may be misleading when e.g. variance of the disturbance is state-dependent. Then, we consider a slightly more general version of (1)

$$\begin{aligned} x_t &\sim p(x_t \mid x_{t-1}) \\ y_t &\sim p(y_t \mid x_t) \end{aligned} \qquad (2)$$

Where $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ denotes probability density function *pdf* of random variable $\boldsymbol{x}_t$ given realization of $\boldsymbol{x}_{t-1}$. Model (1) arises as a special case (2) for choice $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})=N(b(\boldsymbol{x}_{t-1}),Q_t)$ and $p(\boldsymbol{y}_t|\boldsymbol{x}_t)=N(h(\boldsymbol{x}_t),R_t)$. The recursion starts at $t$=0 for $\boldsymbol{x}_0 \sim p(\boldsymbol{x}_0)$ which is known as *prior pdf*.

Model (2) enforces too strong restrictions: (i) realization of state variable $\boldsymbol{x}$ at time t depends only on values of $\boldsymbol{x}_{t-1}$, and (ii) realization of the measurement $\boldsymbol{y}_t$ depends only on current realization of the state $\boldsymbol{x}_t$. These assumptions may seem very restrictive, however, wide range of different models can be converted into the form (2) under appropriate choice of state variable $\boldsymbol{x}_t$. For example, when initial conditions of the process or time-invariant parameters of the *pdfs* are not known, they are considered to be part of the state. In that case, $\boldsymbol{x}_t$ is sometimes called the augmented state, however, we will not make such distinction. In this paper, $\boldsymbol{x}_t$ denotes aggregation of all uncertainty in the model. Specific meaning of different parts of the state will be discussed later.

State-space formulation has been used in DA problem in the later stages of accident in post-emergency phases. Long term evolution of $^{137}$Cs deposited on terrain was predicted recursively (Hofman et al. 2008a) using Kalman filter technique, which is an optimal estimator for linear functions *g*() and *h*() and Gaussian *pdfs* in (2). But such linear model is insufficient for formulation of more complicated problems arising in the early phase of accident (Rojas-Palma 2005) and more general nonlinear dynamic model (2) is required. Bayesian approach to estimation of unknown quantities $\boldsymbol{x}_t$ is based on recursive evaluation of posterior density $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ using the Bayes rule:

$$\begin{aligned} p(x_t\,|y_{1:t}\,) &\propto p(y_t\,|x_t\,)\,p(x_t\,|\,y_{1:t-1}) \\ p(x_{t+1}\,|\,y_{1:t}) &= \int p(x_{t+1}\,|\,x_t)p(x_t\,|\,y_{1:t})dx_t. \end{aligned} \qquad (3)$$

Here, $\boldsymbol{y}_{1:t} = [\boldsymbol{y}_1,\ldots,\boldsymbol{y}_t]$ and $\propto$ denotes equality up to multiplicative constant, see (Ducet et al. 2001) for details. Note that since $\boldsymbol{x}_t$ aggregates all uncertainty in the model, posterior density $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ potentially provides estimates of unknown parameters, unknown initial conditions, or --- under appropriate parameterization --- even unknown variants of the model.

## PARTICLE FILTERING

Except for few special cases (such as the Kalman filter), integration (3) is intractable. Therefore, various approximation has been proposed. The particle filter (also known as sequential Monte Carlo) is based on approximation of the posterior density by a weighted empirical approximation

$$p(x_t\,|\,y_{1:t}) \approx \sum_{i=1}^{n} w_{i,t}\delta(x_t - x_t^{(i)}) \qquad (4)$$

where $x_t^{(i)}$, i=1,…,n are samples of the random variable, i.e. the particles, and $w_{i,t}$ >0, $\sum_{i=1}^{n} w_{i,t} = 1$ are particle weights. Under this approximation, integration (3) is reduced to sampling from densities (in our case $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ ), and recursive evaluation of particle weights $w_{i,t}$.

$$w_{i,t} \propto p(y_t\,|\,x_t^{(i)})w_{i,t-1} \qquad (5)$$

Key advantages of this approximation are easy evaluation of an arbitrary moment, $m(x_t)$,

$$m(x_t) = \sum_{i=1}^{n} w_{i,t}m(x_t^{(i)}) \qquad (6)$$

ability to handle arbitrary non-linear functions, and guaranteed convergence to the true posterior with growing number of particles *n*. The main disadvantage of the approach is its excessive computational cost.

### Adaptation of particle filtering scheme to the early phase of the plume propagation

Intuitively, the key state variable of the scenario is distribution of the pollutant in the atmosphere over the terrain. We model this distribution via segmented Gaussian plume model (SGPM). This is a discrete model with one-hour time step. Within each hour, given amount of a pollutant is released and evolution of this quantity is simulated taking into account all environmental effects (Pecha et al. 2007).

Real release dynamics is partitioned into a number of fictive one-hour segments with equivalent homogenous averaged release source strength. Synchronization with hourly forecast of meteorological conditions is performed. Hourly segment of release is spread during the first hour as a "Gaussian droplet". In the following hours of spreading according to available hourly meteorological forecast, the droplet is treated as "prolonged puff" and its dispersion and depletion during the movement is simulated numerically by large number of elemental shifts. More detailed description of the procedure is described in (Pecha et al. 2008, Hofman et al. 2008). Each hourly segment *g* is consecutively modelled in its all hourly meteorological phases *f*. Output vector $\boldsymbol{s}_T$ of values of interest at time *T* after the release start are superposed as:

$$s_T = \sum_{(g=1)}^{G} \left\{ \sum_{f=g}^{f=T} s^{g,f} \right\} \qquad (7)$$

Each plume segment is uniquely described by the vector variable $\boldsymbol{s}^{g,f}$. Evolution of each such plume segment over the terrain is described by deterministic SGPM model mentioned above. Let rewrite symbolically $\boldsymbol{s}^{g,f}$ to $\boldsymbol{s}(\tau)_t$, where $\tau < t$ denotes time of the release of the plume segment. The SGPM model contains many input and model parameters (Pecha et al. 2005). Most of them are treated as single values that enter the model by their best estimate values. Important random parameters are selected on basis of sensitivity analysis of the SGPM model and constitute random vector $\boldsymbol{\Theta}$. Independent random components of the vector $\boldsymbol{\Theta}$ are labelled as $\Theta_{m,m=1,\ldots,M}$ . Random samples *i* from *pdf*($\Theta_m$) are marked as $\theta_m^{\,i}$ . The components $\Theta_m$ selected for our scenario demonstrates Table 1. The aim of investigations calls for inclusion of as large as possible number of random parameters M. So far, because of computational practicability, the Table 1 presents the case with M truncated to 10.

Variable $\boldsymbol{s}(\tau)_t$ is now parameterized by vector of parameters $\boldsymbol{\Theta}^t$. This vector contains both time invariant parameters, such are dispersion and dry deposition characteristics, and time-variant parameters, such are wind direction and wind velocity at time *t*.

Under probabilistic formalization (2), the original SGPM model is interpreted as conditional density

$$p(s(\tau)_t \mid s(\tau)_{t-1}, \Theta') = \delta\big(s(\tau)_t - SGPM(s(\tau)_{t-1}, \Theta')\big) \tag{8}$$

Parameters $\boldsymbol{\Theta}^t$ were considered to be known in the original formulation. In this text, we consider them to be unknown, hence we consider them to be part of the state. The state is then $x_t = [s(1)_t, s(2)_t, ..., \Theta']$ and its evolution model

$$p(x_t \mid x_{t-1}) = \prod_{\tau=1}^{t} p(s(\tau)_t \mid s(\tau)_{t-1}, \Theta') p(\Theta') \tag{9}$$

Distribution of the parameter vector $p(\boldsymbol{\Theta}^t)$ is composed of independent *pdfs* of components $\Theta_m^t$ given in Table 1.

Table 1: Components $\Theta_m$ of random parameter vector $\boldsymbol{\Theta}$. Count of components truncated to M=10.

| *random parameter* | *unit* | *implementation in code* | *uncertainty bounds* |
|---|---|---|---|
| $\Theta_1$: radioactiv. release during hour 1 (f=1) | [Bq.h$^{-1}$] | $Q = c_1 \times Q^b$<br>$Q^b$ in f=1 | LU; c1∈ <0.31;3.1> |
| $\Theta_2$ : horizont. dispersion | [m] | $\sigma_y = c_2 \times \sigma_y^b$ | $N_{trunc}$ ; c2∈ <0.89;1.12> |
| $\Theta_3$ : dry depo velocity | [m.s$^{-1}$] | $vg = c_3 \times vg^b$ | LU ; c3∈ <0.91;1.10> |
| $\Theta_4$ : Wind direction f=1 | [rad] | $\varphi = \varphi^b + \Delta\varphi$,<br>$\Delta\varphi = c_4 \times 2\pi/80$ | U ; c4∈ <-12;+12> |
| $\Theta_5$ : Wind direction f=2 | [rad] | $\varphi = \varphi^b + \Delta\varphi$,<br>$\Delta\varphi = c_5 \times 2\pi/80$ | U ; c5∈ <-12;+12> |
| $\Theta_6$ : Wind direction f=3 | [rad] | $\varphi = \varphi^b + \Delta\varphi$,<br>$\Delta\varphi = c_6 \times 2\pi/80$ | U ; c6∈ <-12;+12> |
| $\Theta_7$ : Wind speed f=1 | [m.s$^{-1}$] | $V_{10} = c_7 \times V^b_{10}$<br>$V^b_{10}$ in f=1 | U ; c7∈ <0.5;3.0> |
| $\Theta_8$ : Wind speed f=2 | [m.s$^{-1}$] | $V_{10} = c_8 \times V^b_{10}$<br>$V^b_{10}$ in f=2 | U ; c8∈ <0.5;3.0> |
| $\Theta_9$ : Wind speed f=3 | [m.s$^{-1}$] | $V_{10} = c_9 \times V^b_{10}$<br>$V^b_{10}$ in f=3 | U ; c9∈ <0.5;3.0> |
| $\Theta_{10}$: radioativ. release during hour 2 (f=2) | [Bq.h$^{-1}$] | $Q = c_{10} \times Q^b$<br>$Q^b$ in f=2 | LU ; c10∈ <0.31;3.1> |

*Index b stands for "best estimate " values;* $V_{10}$ – wind speed at 10 m height; f – phase (hour) after the release start;

Type of distribution: LU-loguniform; $N_{trunc}$ – Normal, truncated; U – Uniform;

The measurements are modelled to have Gaussian distribution:

$$p(y_t \mid x_t) = N\Big(\sum_{\tau=1}^{t} SGPM(s(\tau)_t), \Sigma_t\Big) \tag{10}$$

The mean value is given by the sum of outputs from each plume segments and is approximated by bilinear approximation of the SGPM model predictions at the points of measurements. For the experiment purposes the covariance matrix $\Sigma_t$ is constructed as

$$\Sigma_t = \lambda_{\text{model}} I_M + \lambda_{\text{prop}} \text{diag}(y_t) \tag{11}$$

with chosen constants $\lambda_{\text{model}}$ and $\lambda_{\text{prop}}$. The first term models inaccuracies of the chosen Gaussian plume approximation, the second term models inaccuracies of the measuring devices. This model is almost an arbitrary choice, that is used to show potential of the considered methodology. Model of observation for practical purpose should be designed using exact characteristics of the application specific measurement devices.

## Implementation of PF algorithm

The following steps represent computational flow of recursive particle filtering applied here:

1. Generate *n* realizations of parameter vector $\boldsymbol{\Theta}^0$ from densities listed in Table 1, $\{ [\theta_m^{\ i}]_{m=1:M} \}^{i=1:n}$. Substitution of sets of realisations $[\theta_m^{\ i}]_{m=1:M}$ for each *i* into SGPM model (7) yields *n* corresponding plumes (in the following text interpreted as "particle"). Initially, the same weight $w_{i,0}=1/n$ is assigned to each particle *i*.
2. For each time t=1...T :
    a. Generate a set $\{ [\theta_m^{\ i}]_{m=1:M} \}^{i=1:n}$ of realizations of $\Theta^t$ and for each plume (particle) compute one step prediction using the SGPM algorithm. The term "particle prolongation" is introduced.
    b. If measurements are available, recompute the weights $w_{i,t}$ using (5).
    c. Compute posterior values of parameters of interest using (6)

Parameter vector $\boldsymbol{\Theta}$ is expressed in Equation (8) as $\boldsymbol{\Theta}^t$. It means that count of the components treated as random within a certain time interval can vary, symbolically:

$$\boldsymbol{\Theta}^{t=1} \approx \boxed{\Theta_1 \quad \Theta_2 \quad \Theta_3 \quad \Theta_4} \quad \Theta_5 \quad \Theta_6 \quad \boxed{\Theta_7} \quad \Theta_8 \quad \Theta_9 \quad \Theta_{10} \ldots.$$

$$\boldsymbol{\Theta}^{t=2} \approx \boxed{\Theta_1 \quad \Theta_2 \quad \Theta_3} \quad \Theta_4 \quad \boxed{\Theta_5} \quad \Theta_6 \quad \Theta_7 \quad \boxed{\Theta_8} \quad \Theta_9 \quad \boxed{\Theta_{10}} \ldots.$$

$$\boldsymbol{\Theta}^{t=3} \approx \Theta_1 \quad \Theta_2 \quad \Theta_3 \quad \Theta_4 \quad \Theta_5 \quad \boxed{\Theta_6} \quad \Theta_7 \quad \Theta_8 \quad \boxed{\Theta_9} \quad \boxed{\Theta_{10}} \ldots.$$

Let assume only the first three hours from the same beginning of an accident. It corresponds to 10 parameters from Table 1. Bounded components stand for the relevant components that enter the sampling procedure in the particular time step. Alternative resampling schemes could be constructed (e.g. locally dependant land use characteristics when corresponding $\theta_2$ and $\theta_3$ could be assumed relevant in all time steps).

## Experimental results

The sampling scheme consists of generation of 5000 particles corresponding to *n*=5000 realisations of random parameter vector $\boldsymbol{\Theta}$ with 10 components $\Theta_{m,m=1,\ldots,10}$ according to uncertainty characteristics described in Table 1.

Evaluated values of the particle weights using $\lambda_{\text{model}} = 10^4$ and $\lambda_{\text{prop}} = cov \times \kappa$ , with *cov* =1,…5 , are illustrated in Figure 1. The smallest values of variance (top) sharply selects only a few particles. With increasing variance, *cov*=2, …,5, uncertainty in the weight grows and more particles become non-negligible. Constant κ=1.0E+6 ensures link to measured magnitudes of radioactivity deposition.

Prior and posterior histograms of distributions of some parameters $\Theta_m$ from Table 1 are compared in Figure 2. Note that the posterior is sharply peaked for the three leftmost parameters while it is still widespread for the remaining parameters. But we should distinguish between parameter estimation for the concrete analysed situation and common average conditions. It should not be confused with parameter estimation which could give recommendation on parameter values commonly valid “in average”.

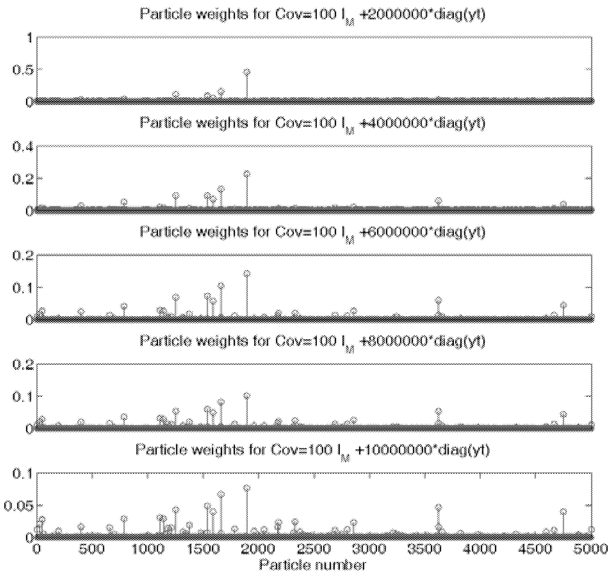


Figure 1: Posterior weights $w_t$ for five choices *cov*, update using measurements incoming just after 2 hours from start.

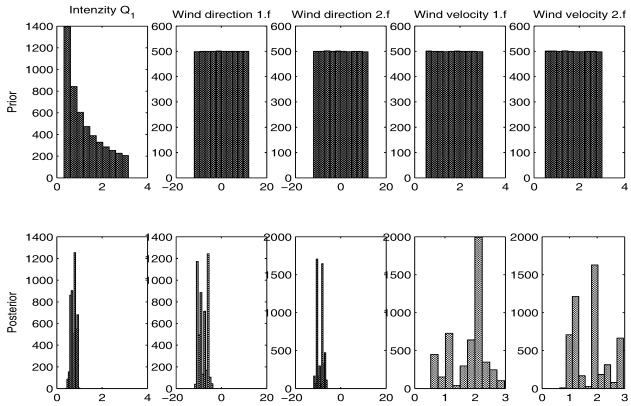


Figure 2: Comparison of prior (top row) and posterior (bottom row) histograms of distribution of selected parameters for *cov*=3.

## ILLUSTRATION OF PARTICLE FILTERING APPLIED IN THE EARLY STAGE OF A HYPOTHETICAL ACCIDENT

The robustness of the PF method outlined above is illustrated for case of a certain circumstance when in the same beginning of an accident the decision maker is not provided by fully clear and unambiguous information. Experience from former radiation accidents pointed out the side effects leading to an information shocks with possible temporal paralysis of communication lines. In this sense we have adjusted a hypothetical accident scenario. Real meteorological situation from March 31, 2009 is taken into consideration and the moment of hypothetical radioactivity release is set to 10.00 CET. Available real meteorological observations measured at the point of nuclear power plant (NPP) and short term meteorological forecast are somewhat inconsistent (see next Table 2). Following ex post analysis can give a retrospective view on such the atypical situations ( their occurrence rate is surprisingly not negligible). Due to a possible information shock mentioned above we shall assume conservatively a delay of two hours in recovery of radiation monitoring. Thus, the first measurements from terrain are coming just two hours after the release started. A decision maker has a dilemma how to manage the prediction of harmful substances in the early stage.

### Available meteorological data

Let release of $^{131}$I radioactivity has started at 10.00 CET, March 31, 2009, and lasted for 2 hours (Table 2).

Table 2: Accidental release scenario of $^{131}$I , short-term meteorological forecast and real meteorological measurements for “point” of NPP Temelin ( 49°10'48.53"N × 14°22'30.93"E), time stamp 20090331-1000 CET.

| CET hour | 10.00 | 11.00 | 12.00 | 13.00 |
|---|---|---|---|---|
| activity release of $^{131}$I Bq/hour | 5.68 × e+14 | 7.92 × e+14 | 0 | 0 |
| wind direction$^1$ **METLOC/METOBS** | 95.0 / 54.0 | 101.0 / 69.0 | 84.0 / 65.0 | 80.0 / 80.0 |
| wind speed$^2$ **METLOC/METOBS** | 2.0 / 3.8 | 2.1 / 3.0 | 1.9 / 3.8 | 2.2 / 3.8 |
| Pasquill atm. stability | A | A | B | B |

*$^{1)}$ ... at 10 m height, blowing “from” (degrees measured clockwise from North); $^{2)}$ ... at 10 m height (m/s)*

At moment of accident, the three kinds of meteorological data were directly available:

- Short term meteorological forecast generated twice a day, sequences up to 48 hours):
  - Label METLOC: Simple local forecast for the point of NPP (hourly sequences of wind direction and speed, category of atmospheric stability according to Pasquill and precipitation),
  - Label METGRID: 3-D meteorological forecast in HIRLAM format for vicinity 160 × 160 kilometers around NPP.
- Label METOBS: Observed values (real online meteorological measurements) incoming automatically from the point of NPP.

All the data are provided by the Czech meteorological service and are available online through ORACLE DB server.

Deterministic calculations according to SGPM model with METLOC meteorology for the first two hours of the release are illustrated in Figure 3. Superposition according to Equation (7) was used for quantity of $^{131}$I deposition on the ground (first segment g=1, in phases f=1 and 2; second segment g=2, in phase f=2 ).

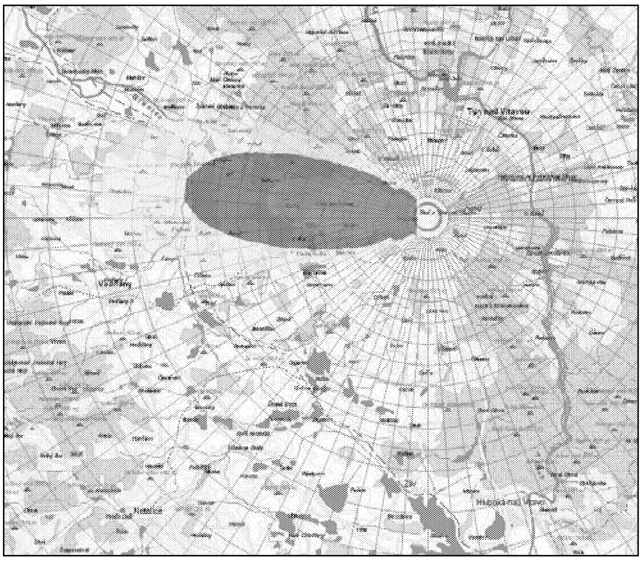


Figure 3: Release scenario with meteodata METLOC - model predictions for "best estimate" values of model parameters, just 2 hours after the release start.
$^{131}$I deposition ranges (Bq.m$^{-2}$): red: 5.00e+06 ÷ 1.30e+08 ; blue: 1.00e+06 ÷ 5.00e+06 ; yellow: 1.00e+05 ÷ 1.00e+06 ;

**Arrangement of the real positions of monitoring sensors**

Early Warning Network (EWN) such a component of existing Radiation Monitoring Network (RMN) of the Czech Republic can be exploited for purposes of DA procedures. The main part of EWN is teledosimetric system (TDS) which for the NPP Temelin consists of two circles. The inner circle is positioned on the NPP-fence (see red circles in Figure 4 very close to NPP or in better discrimination in Figure 5) and consists from 24 stations 2,5m above ground. The outer II. circle of measurement positions is drown in Figure 4 by red squares. The dose-rate data are transferred each 4 minutes and stored to the ORACLE DB server for online access. We are assuming all these receptors to be operable. An ability to measure selected magnitudes of deposition is a question of a future monitoring development.

For DA purposes we have 79 sensors located in vicinity of the nuclear facility. In this number we have included 3 mobile stations located randomly in the middle distances.

**Artificial simulation of the missing real accidental radiological data**

We hope that all considerations remain only in hypothetical level and that the testing accidental radiological data will be always generated artificially. The technique is sometimes known as "twin experiment".

A degree of belief to the initial near-range estimation using the SGPM model predictions with METLOC meteorological forecast (see Figure 3) will be low if we take into considerations the similar calculations with METOBS real meteorological measurement (see Figure 4). We should respect the fact that if something happens, the shape of the corresponding accidental trajectory close to the source should correspond more likely with the Figure 4.

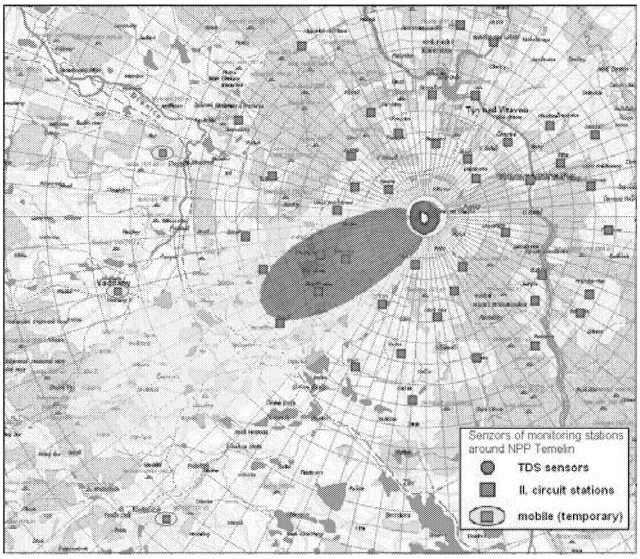


Figure 4: Release scenario with meteodata METOBS - model predictions for "best estimate" values of model parameters, just 2 hours after the release start.
$^{131}$*I deposition ranges (Bq.m$^{-2}$)*: the same as in Figure 3. This figure also illustrates configuration of the inner part of the Czech EWN around NPP Temelin.

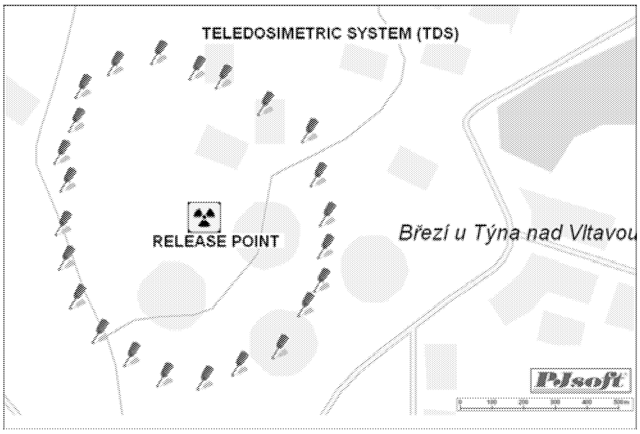


Figure 5: TDS on fence of NPP Temelin – 24 detectors

Without more discussion, we use this subjective assumption and generate the "artificial measurements" on the basis of METOBS real meteorological measurement in Figure 4. Though the model requirements so far exceed possibilities of monitoring in the Czech Republic, the cooperation between modelers and monitoring has growing importance. The assimilation subsystem is developed in cooperation with National Radiation Protection Institute (NRPI) which is administrator of RMN.

## THE RESULTS ACHIEVED FOR SEVERAL FIRST TIME STEPS

Finally, the following hypothetical data assimilation scenario defined for the early phase is accomplished:

1. Predictions of 5000 particles (trajectories) for 5000 realisations of the parameter vector **Θ** according to the SGPM model. It covers time interval 2 hours from the same beginning of accident, no measurements from terrain are not yet available. Gridded meteorological forecast METGRID

is always used. Prior probabilistic density function and its moments can be estimated.

2. Let the first set of "artificial measurements" is incoming just two hours after the release start. The values of "measurements" are generated according to Figure 4 and the speculations introduced above.
3. Update step of recursive procedure of PF estimates the posterior density function on basis of weighted empirical approximation given by Equation (4).
4. Recursion continues in the next time interval performing the transition step with resampled particles.

Approximation of posterior *pdf* is generated for 5 choices of covariances *cov*=1,….,5 according to Equation (11) (where $\lambda_{\text{prop}} = cov \times \kappa$ ) and "measurements" from Figure 4. Expected mean values are calculated using common expression according to Equation (6), specifically in the form:

$$I(f_t) := E_{p(x_t|y_{1:t})}\left[f(x_t)\right] = \int f(x_t)\, p(x_t \mid y_{1:t})\, dx_t \qquad (12)$$

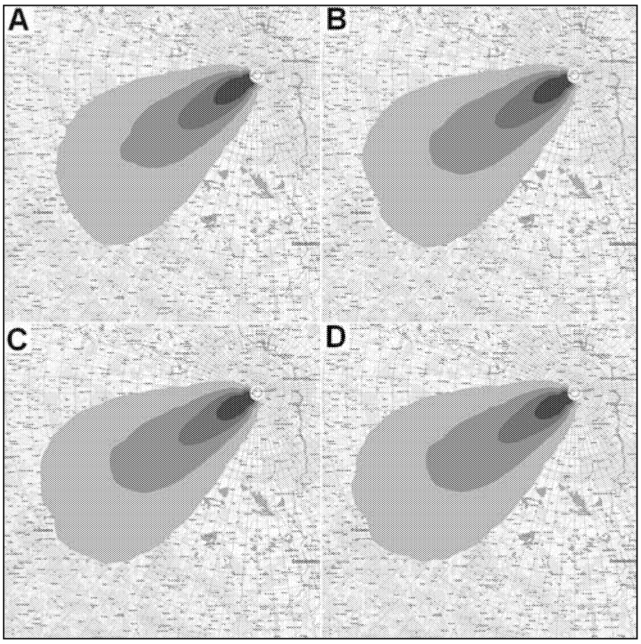


Figure 6: Expectations of posterior *pdf* of the radioactivity deposition in dependency on covariance matrix (according to Equation (11) ). A,B,C,D stand for *cov*=1,2,4,5.

An estimation of the expectations on basis of *n* generated particles $x_t^{(i)}$ , *i=1:n* from posterior distribution is given by:

$$I^n(f_t) = \frac{1}{n}\sum_{i=1}^{n} f\left(x_t^{(i)}\right) \qquad (13)$$

For *n*→∞ is achieved almost sure convergence of $I^n(f)$ to I(f).

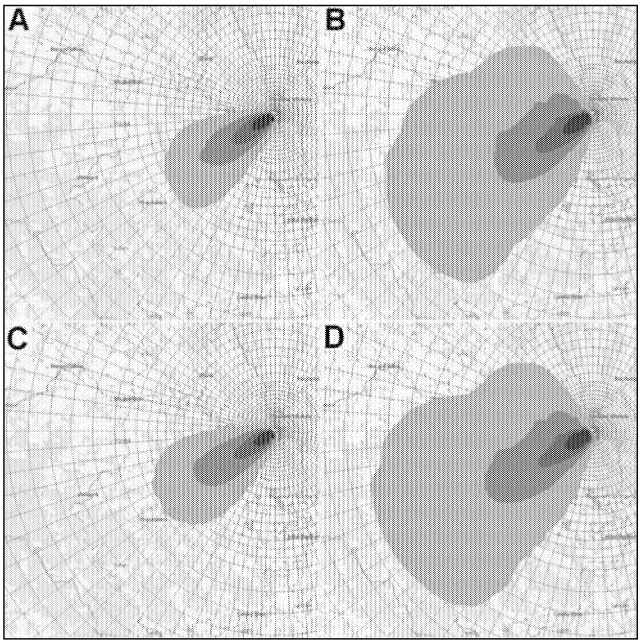


Figure 7: Transition step for the next time interval. Prior *pdf* expectations for transition from hour 2 to hour 3. (case A → B for *cov*=1; case C → D for *cov*=5).

The expectations of the quantity of activity deposition are given in Figure 6 for cases of *cov*=1,2,4,5. The outer contour corresponds to the level of 1.00 E+03 Bq.m$^{-2}$. The results show tendency of the updated model to approach the measurements with low noises. The values are slightly spreading when inaccuracies of measurements grows (higher *cov* ). Covariances of the measurement errors were selected rather low. At present new tests with increased covariance are running and tendency to lean to either model predictions or measurements are mapping.

Figure 7 demonstrates prolongation one time step forward. Case A concerns *cov*=1 (also in Figure 6 A) expectation from the posterior density just after 2 hours after the release start. Numerical approximation of the SGPM model is used for solution of the second part of Equation (3) which stands for transition equation for specific formulation $p(\boldsymbol{x}_{f=3} \mid \boldsymbol{x}_{f=2})$. Prediction from analysis (data update) in the second hour (upper left A) to the third hour (upper right B) is done (prediction step). SGPM model prolongs the weighted particles within the hour 2 → 3. The similar shift for *cov*=5 stands for cases C → D.

## CONCLUSION

The article extends former investigations in DA methodology (Hofman et al. 2007) where analysis of the input model parameters uncertainty and both model error and observation error covariance structure were examined. DA in early stage of accident requires much more sophisticated access. From all possible techniques is adopted particle filter, which has one significant attribute. In PF the state ensemble trajectories are kept unchanged during the update step as for the forecast step and only their weights are updated. The particles remain unchanged after the correction (update) step and only receive the new weight ( according to Equation (5) ) reflecting closeness of the particle with respect the new observations.

This evident PF feature has favourable impact on exploitation of nonlinear prediction model SGPM in DA process in the early stage. SGPM model is in principle a trajectory model. The PF does not disrupt the trajectory information and it can be easily recursively forwarded.

The presented approach brings advantage of fast computation even for large number of realisations. One PF step of update and predictions with 5000 realisations is accomplished during about 15 minutes (common PC config.) and promises to support the decision making process in real time.

The adopted procedure seems to be robust and suitable to manage a certain discrepancies and scenario incompletness occurring from the same beginning of an accident. The authors narrow down anxiously the range of some uncertainties. For example the range of horizontal dispersion uncertainty $c_2$ and dry deposition $c_3$ should be much higher (in correspondence with expert judgments). Afterwards, the traces (e.g. in Figure 6) would be more dispersed in horizontal and longitudinal directions. Even the calculations have covered only the first time step and demonstrated code ability to predict in the second step, the full recursive PF application seems to be easily feasible.

Still open remains a question of availability of measurements, capability to provide specific quantities and configuration and density of monitoring stations. The first negotiation between modellers and specialists responsible for monitoring was launched (Kuca et al. 2008). The poor information can result from rare measurements. On the other hand, requirements issued from DA experience should be reflected in the future development of radiation monitoring networks.

DA plays substantial role in realistic prediction of evolution of radiation situation during nuclear emergency. Reliable information arriving on time provides decision makers with necessary time on judgement and introduction of efficient urgent countermeasures on population protection.

## ACKNOWLEDGEMENTS

This work is part of the grant project GAČR No. 02/07/1596, which is founded by Grant Agency of the Czech Republic. The activities has been also supported by Research center DAR. Thanks to National Radiation Protection Institute in Prague, the online connection of developed software to meteorological forecast and radiological data from Early Warning Network of the Czech Republic could be tested.

## REFERENCES

Arulampalam, M. S.; S. Maskell, N. Gordon and T. Clapp. 2002. "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking". *IEEE Transactions on Signal Processing, Vol. 50, No. 2.*

Drécourt, J-P. 2004. "Data assimilation in hydrological modeling". Environmental&Resources, Techn. Univ. of Denmark, ISBN 87-89220-84-6.

Doucet, A.; N. De Freitas and N.J. Gordon (eds.). 2001. S*equential Monte Carlo method in practice.* Springer-Verlag, New York.

Doucet, A. and A.M. Johansen. 2009. "A Tutorial on Particle Filtering and Smoothing: Fifteen years later". *Will be published in Handbook of Nonlinear Filtering (2009).*

Moradkhani, H. 2008. "Hydrologic Remote Sensing and Land Surface Data Assimilation". *Sensors, 8, 2986-3004, ISSN 1424-8220.*

Hofman, R. and P. Pecha 2007. "Integration of data assimilation subsystem into environmental model of harmful substances propagation". In *Proc. of the 11$^{-th}$ Int. Conf. on Harmonisation within Atmospheric Dispersion Modelling* (Cambridge, UK, July 2-5, 2007), 111-115.

Hofman, R.; P. Pecha ; and E. Pechová . 2008. "A simplified approach for solution of time update problem during toxic waste spreading in the atmosphere", *Hrvatski Meteorološki Časopis, 12$^{-th}$ Int. Conf. on Harmonisation within Atmospheric Dispersion Modelling* (Cavtat, HR, Oct. 6-10, 2008) .

Hofman, R. and P. Pecha 2008a „Data Assimilation of Model Predictions of Long-Time Evolution of $^{137}$Cs Deposition on Terrain" , Proceeding of the IEEE International Geoscience & Remote Sensing Symposium 2008, IEEE International Geoscience & Remote Sensing Symposium 2008, (Boston, US, 06.07.2008-11.07.2008) (2008).

Hoteit, I., D.-T. Pham, G. Triantafyllou and G. Korres. 2008. Particle Kalman Filtering for Data Assimilation in Meteorology and Oceanography. *Mon. Wea. Rev., 136.*

Kalnay, E. 2003. *Atmospheric modeling, data assimilation and predictability*. Cambridge University Press, ISBN 0-521-79179-0.

Kuca, P., R. Hofman and P. Pecha. 2008. "Assimilation techniques in consequence assessment of accidental radioactivity releases – the way for increase of reliability of predictions". In. *Proc. of ECORAD 2008 Int. Conf. On Radioecology & Environ.Radioactivity* (Bergen, Norway, June 15-20, 2008), 138-141, ISBN 978-82-90362-25-1.

Pecha, P., R. Hofman and E. Pechova. 2007. "*Traini*ng simulator for analysis of environmental consequences of accidental radioactivity releases". In *Proc. of the 6$^{th}$ EUROSIM Congress on Modelling and Simulation* (Ljubljana, Slovenia, Sept. 9-13, 2007), 18 pp., CD– ISBN 978-3-901608-32-2.

Pecha, P. and E. Pechova. 2005. "Modeling of random activity concentration fields in air … ". *In Proc. of the 10$^{-th}$ Int. Conf.. on Harmonisation within Atmospheric Dispersion Modelling* (Sissi, Greece, ), paper No. H11-069.

Pecha, P. and R. Hofman. 2008. "Fitting of segmented Gaussian plume model predictions on measured data". *In Proc. of the 22th European Simulation and Modelling Conference ESM'2008* (LeHavre, FR, Oct 27-29,2008)

Pecha, P., R. Hofman and V. Šmídl. 2009. "Simulation of random 3-D trajectories of the toxic plume spreading over the terrain". In *Proc. of the 5$^{th}$ WMODA conf on Data Assimilation (Melbourne, Australia, Oct. 5-9, 2009).*

Rojas-Palma, C. 2005, Data assimilation for of site nuclear emergency management. Technical report, SCK-CEN, DAONEM, RODOS(RA5)- final report, RODOS(RA5)-RE(04)-01, 2005.

# COUPLED ATMOSPHERE-WILDFIRE MODEL FOR PILOTED FLIGHT SIMULATION OF AERIAL FIRE FIGHTING OPERATIONS

Luca Cistriani
UAS & Simulators Business Area
SELEX GALILEO
Via M. Stoppani, 21
34077 Ronchi dei Legionari (GO),
ITALY

Sebastiano Bonfiglio
Aerospace & Defence Department
ALTRAN Italia
Via Flavia, 23/1
34148 Trieste,
ITALY

Nicola Stella
Remote Sensing Area
GEOTEC S.R.L.
Via Collodi, 5B
75100 Matera
ITALY

**KEYWORDS**
Flight Simulation, Aerial Firefighting, Coupled Map Lattice, Rothermel, Cellular Automata, Matlab Simulink, Real Time Workshop.

## ABSTRACT

A computational model of wildfires has been developed to improve the functionalities of a flight training device developed for the operational training of CL-415 water bomber flight crews. The model, realized in Matlab Simulink, is based on a coupled atmosphere-fire model that is able to simulate the rising plume introduced in the wind pattern by the wildfire depending on the overall wind, temperature, pressure and humidity conditions set by the instructor in terms of ISA and non-ISA conditions in conjunction with MIL-F-8785C windshear model. Actual fire spread and suppression are simulated through the implementation of a cellular automata model fed by a GIS system, while the Coupled Map Lattice (CML) method is used to simulate wind and temperature profiles in the atmosphere block surrounding the wildfire.

Some simplifications have been accepted in order to speed-up the simulation that is run in real-time as a part of the synthetic environment embedded in the flight training device. Good qualitative results have been obtained despite of the simplifications assumed, providing an effective tool to improve the crew perception of fire-related hazards during aerial fire-fighting operations.

## INTRODUCTION

Millions of hectares of forestry are destroyed by fire worldwide each year. Firefighting operations are of primary importance in the European Union, especially for those countries facing the Mediterranean Sea, where the combination of densely populated areas, soils characterized by a brush vegetation and low moisture content, elevated temperatures and strong winds during summer cause an ideal scenario for the ignition and propagation of forest fires. As an example, 7,797 fires have been registered only in Italy during the year 2007, with 127,151 hectares of landscape burnt, 61,100 of which being forest (Corpo Forestale dello Stato 2007).

Due to the severity of the threat, aerial firefighting operations represent a significant effort for these countries especially during the summer season, pushing the need for an efficient fleet of large aerial firefighting aircraft in order to guarantee a significant water delivery capability in short time.

As a consequence of this need, several countries employ many different fixed and rotary wing aerial assets with Italy, France, Greece, Croatia and Spain being the major European operators of a well-known amphibious aircraft type specifically designed for the mission of aerial firefighting, namely the Bombardier Model CL-215-6B11 also known as Canadair CL-415.

Figure 1: Canadair CL-415 Operated by SOREM for the Italian Protezione Civile.

Totally, the number of aircraft operated by the five countries in Europe is about 54 (Jane's 2008) but is actually increasing due to several orders on delivery; Italy is actually the major operator with 22 aircraft in active service.

Flying a fire-fighting mission is a hazardous task since many threats are present in the scenario: smoke limits visibility and may cause an engine flame-out or, even worst, an engine fire due to burning leafs and sparks lifted up by the fire-induced buoyancy. If this is not the case, a sensible decay of performance can be expected due to the lower air density in the buoyant plume. Local winds and turbulence may be caused also by the proximity with the terrain (water is delivered usually at 100 ft above ground), that inevitably forces the pilots to fly across strong wind corridors, rotors or other forms of local winds induced by the orography of the terrain.

Due to the above, high skill levels are required to pilots operating water bombers, while at the same time obvious safety considerations preclude effective in-flight training for specific emergencies (like an engine failure during take-off, water pickup or final water bombing run).

Despite the high number of aircraft and operators involved, actually there are no CL-415 Flight Simulators available in Europe. To fill this gap, Selex-Galileo, with the support of the Regione Autonoma Friuli Venezia Giulia and in partnership with Italian Protezione Civile and SOREM (the company that operates the Italian Canadairs) has developed a demonstrator for a full mission flight simulator specifically devoted to firefighting. The Demonstrator developed so far incorporates a simulation model of the CL-415 aircraft, and a complex simulation of the environment, able to reproduce in real-time the behavior of a forest fire and its associated meteorological phenomena relevant to flight mechanics: local winds, bursts, temperature gradients etc. The simulation of the environment is interactive with the aircraft missions in two ways: the aircraft behaviour is influenced by the perturbations introduced in the atmosphere by the wildfire and the wildfire itself is influenced by the aircraft firefighting action (dropping of water) which affects fire spread on the ground.

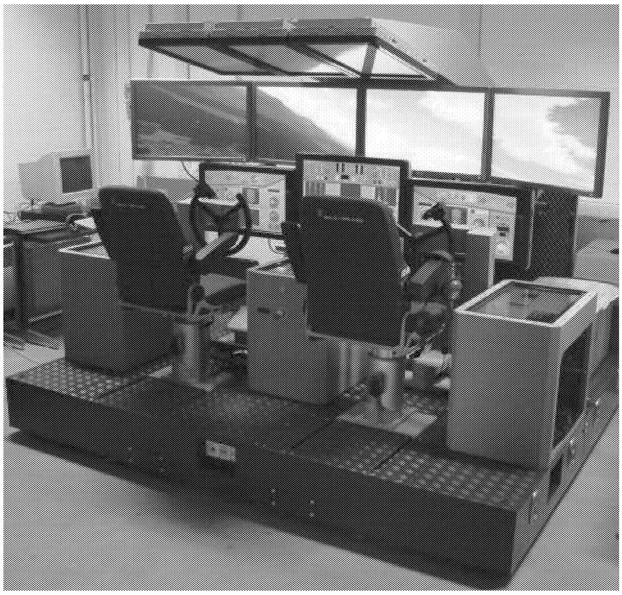


Figure 2: Mission Simulator for theCanadair CL-415 developed at Selex-Galileo.

## REPRESENTATION OF DISTURBANCES IN FLIGHT SIMULATION

The Flight Simulator is composed by a low fidelity replica of the twin-seat CL-415 cockpit (see Fig. 2). The aircraft instrumentation is emulated on nine touch-screens reproducing the overhead, front and side panels of the aircraft. Handweels (with associated triggers and buttons), pedals, power and propeller throttles and water pickup probes are physically reproduced with an hardware replica of the actual device. Loads on the primary flight controls are reproduced through an electrical control loading system. Aircraft and ambient sounds are reproduced but a motion system is not present. The simulator allows normal aircraft handling, including all normal mission checklists and a limited number of emergencies. Environmental effects are grouped in two major blocks:

- General purpose atmospheric disturbances
- Fire induced atmospheric disturbances.

The first group includes all the atmospheric disturbances generally present in a simulator of this kind modeled according to MIL-F-8785C: steady wind with three different vertical profiles (constant wind, low and high altitude windshear model), atmospheric turbulence (McFarland 1997) is modeled using Dryden spectra tunable in intensity (Light, Moderate, Severe) and type (low and high altitude model); isolated bursts and microbursts models are included.

The second group includes all those disturbances that can be expected in the vicinity of a wildfire and that constitute the specificity of such a simulator. The effect of disturbances is simulated in the aircraft flight mechanics by adding wind-induced translational and rotational velocities to the actual body rates and velocities.

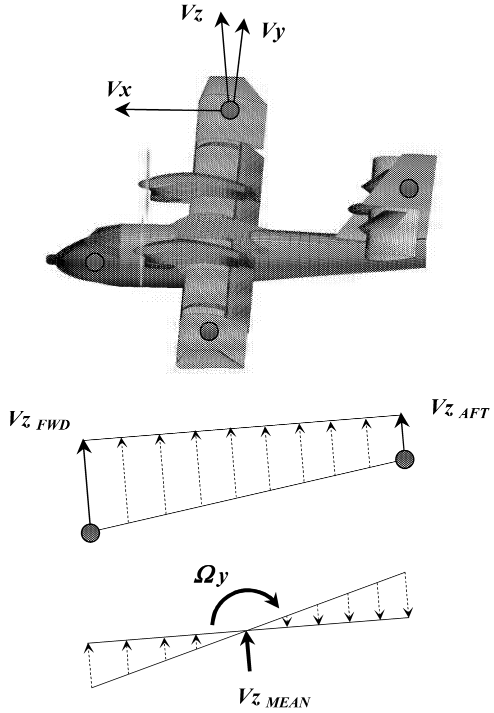


Figure 3: Sampling points for the evaluating wind velocities and decomposition into equivalent mean value and body rate.

The representation of fire-induced disturbances on aircraft behavior is reduced to the calculation of wind velocities along the aircraft flight path at the four evaluation points (See Fig. 3). This is accomplished by interpolation of wind velocities over a spatial grid covering a square domain surrounding the wildfire area; wind velocities are updated in real-time according to the evolution of the coupled fire-atmosphere model.

## COUPLED WILDFIRE-ATMOSPHERE MODEL

The idea at the basis of a coupled fire-atmosphere model is to catch the main interactions between the two physical phenomena that are:

- Fire releases heat in the atmosphere causing a buoyant plume; mass conservation produces diffused areas of entrainement at the plume edges.
- The wind pattern on ground is influenced by the areas of entrainment, thus influencing the fire spread.

Coupled wildfire-atmosphere models are often employed in simulations devoted to emergency management (Coen 2003, 2005 and Coen et al. 2006); generally the approach is to couple a fire spread algorithm like FARSITE or BEHAVE to an atmosphere simulation package of the same type of those commonly used for meteorological forecasts like WRF (Patton & Coen 2004).
Figure 4 illustrates schematically the interactions between the aircraft, fire and atmosphere models. Pseudocolors indicates a wind speed in excess (red) or defect (blue) of the far field condition.

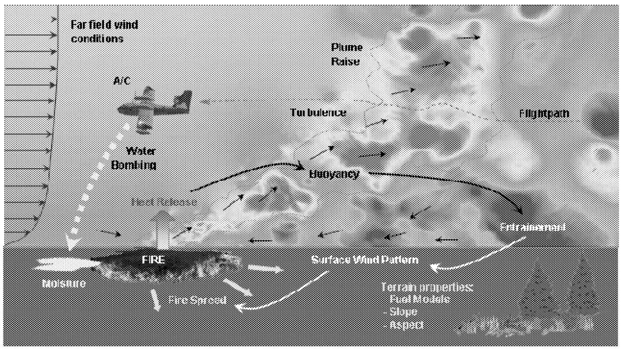


Figure 4: Conceptual sketch illustrating Fire-Atmosphere-Aircraft mutual influences.

The peculiarities of the application to a flight simulation of an aerial firefighter are:

- The spatial discretization of the atmosphere computational grid should be such that small scale phenomena can be resolved to allow proper simulation of aircraft motion through the disturbances.
- The spatial resolution of the wildfire computational grid should be such that the firefighting action (pouring water into the fire) can be effectively resolved reproducing the effects of slope, drop altitude, aircraft speed etc.
- The execution times of the fire and atmosphere models should be such that real-time execution can be performed.

Fire spread on the ground is affected by "static" terrain properties (fuel models, slope, aspect) but also by dynamic quantities as the induced wind and the water drops.

### Fire Spread Model

The simulation spread model is based on the implementation of cellular automata algorithm using a set of mathematical equations developed in C language (derived from the BEHAVE model) and imported in the Simulink environment. BEHAVE is a fire behavior prediction system developed in early 80's. It allows predicting a set of fire features such as rate of spread and fire intensity using 13 types of terrain model and different weather conditions as required in the Rothermel model (Rothermel 1972).
The fire behavior is modeled starting from two-dimensional square grids derived from a GIS system: it can be represented as several different layers where each layer holds data about a particular kind of feature.

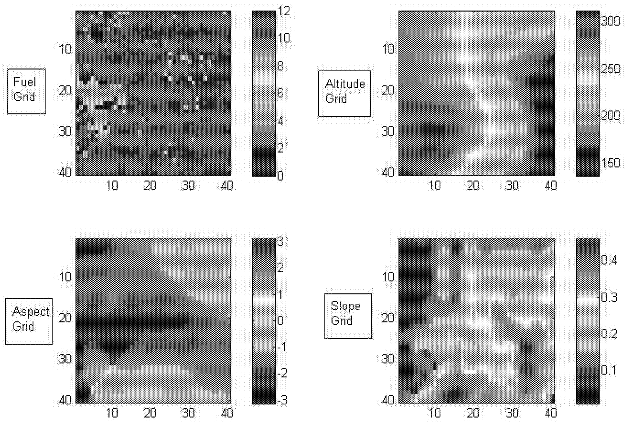


Figure 5: 40 x 40 grid types obtained from GIS data.

In particular, the simulation model requires as input:

- Fuel Models
- Terrain Slope
- Terrain Aspect

Fuel Models consist of 13 different type of terrain, used by Albini (1976) including 11 developed by Anderson and Brown and published by Rothermel (1972).
Slope is the change in elevation with respect to change in horizontal position. In our case, slope gives the maximum rate of change between each cell and its neighbors.
Aspect is the direction of the plane with respect to some arbitrary zero (north). The value of each cell indicates the direction the cell's slope faces.
User can choose the initial ignition point/points of the considered map by setting an input properly. Fire algorithm will consider this point/points as burning starting from next simulation time step. One or more ignition points can be added during simulation.
From these data it is possible to calculate the time required for a fire to travel from a burning cell to one of the adjacent eight cells under particular weather conditions (moisture, wind speed and wind direction). Fire propagation from cell to neighbor cells is computed by Rothermel equations according to the speed and direction of the local wind,

remaining fuel in the cells and elapsed time since the fire broke out.

The model also takes care of simulation bonds such as time-step and a-synchronous inputs (eventually) coming from other systems, such as moisture growing effects due to water bombing actions.

At each simulation step, for each cell an ignition time is assigned, depending on the previous step simulation inputs. The algorithm loops on the entire grid: a cell starts burning when its Ignition Time is equal to ActualTime. If a cell is burning, eight ignition time values (corresponding to the eight neighbors) are provided eventually updating the previous ones. The TimeNext value (updated at the end of the loop) represents the next burning cell time and permits to monitor the correct simulation sequence.

See Figure 7 for an example of burning cell sequence under constant environment condition.

The flow chart of the spread algorithm is shown in Figure 6.

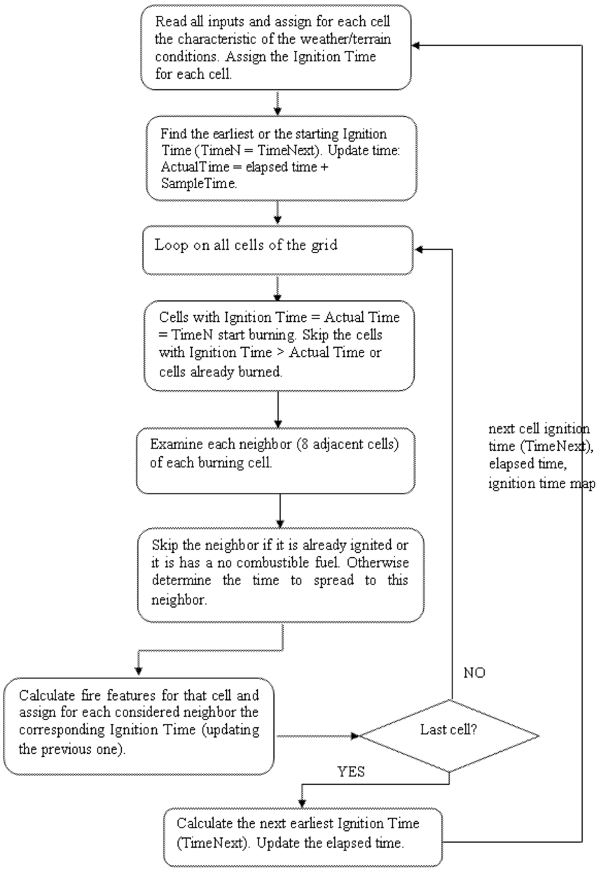
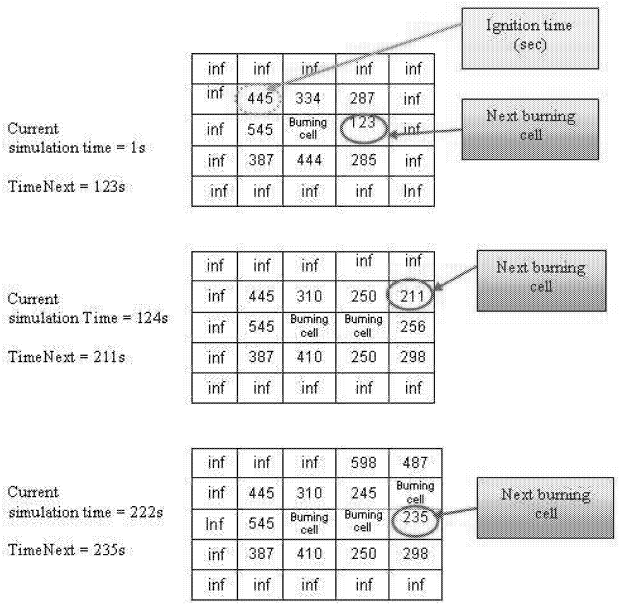

Read all inputs and assign for each cell the characteristic of the weather/terrain conditions. Assign the Ignition Time for each cell.
Find the earliest or the starting Ignition Time (TimeN = TimeNext). Update time: ActualTime = elapsed time + SampleTime.
Loop on all cells of the grid
Cells with Ignition Time = Actual Time = TimeN start burning. Skip the cells with Ignition Time > Actual Time or cells already burned.
Examine each neighbor (8 adjacent cells) of each burning cell.
Skip the neighbor if it is already ignited or it is has a no combustible fuel. Otherwise determine the time to spread to this neighbor.
Calculate fire features for that cell and assign for each considered neighbor the corresponding Ignition Time (updating the previous one).
Last cell?
NO
YES
next cell ignition time (TimeNext), elapsed time, ignition time map
Calculate the next earliest Ignition Time (TimeNext). Update the elapsed time.

Figure 6: Flow chart of fire spread algorithm.

Moreover, an interface has been developed in Simulink in order to observe some simulation constraints (for example: the interfacing with other systems) and to avoid that a burned cell could start burning again.

Fire properties (i.e. Reaction Intensity, Flame Height, etc.) are calculated at each simulation step through the C library: it contains BEHAVE fire behavior algorithms, encapsulated and optimized for fire behavior simulation. Access to library is performed via C macros that are used like C function calls to access or update current object properties.



| | | | | |
|---|---|---|---|---|
| inf | inf | inf | inf | inf |
| inf | 445 | 334 | 287 | inf |
| inf | 545 | Burning cell | 123 | inf |
| inf | 387 | 444 | 285 | inf |
| inf | inf | inf | inf | Inf |

| | | | | |
|---|---|---|---|---|
| inf | inf | inf | inf | inf |
| inf | 445 | 310 | 250 | 211 |
| inf | 545 | Burning cell | Burning cell | 256 |
| inf | 387 | 410 | 250 | 298 |
| inf | inf | inf | inf | inf |

| | | | | |
|---|---|---|---|---|
| inf | inf | inf | 598 | 487 |
| inf | 445 | 310 | 245 | Burning cell |
| Inf | 545 | Burning cell | Burning cell | 235 |
| inf | 387 | 410 | 250 | 298 |
| inf | inf | inf | inf | inf |

Figure 7: Cells burning sequence under constant environment conditions.

### Atmosphere Model

The dynamic simulation of the atmosphere in the vicinity of the wildfire is based on the Coupled Map Lattice (CML) method. The CML is a simplified numerical method for the solution of non linear equations originally proposed by Kaneko (Kaneko 1990). An efficient computational method based on CML for the solution of fluid equations was proposed by J. Stam (Stam 1999) with application to the visual simulation of smoke. CML have been used extensively to simulate fluid motions, especially for real-time computer graphics, since its implementation on graphic hardware considerably speeds up the calculation and rendering times (Crane et al.2007, Wu et al. 2004). The implementation we adopted is based on some works related to cloud dynamics (Mizuno et al. 2002, Miyazaki et al. 2002, 2001, Dobashi et al. 2000).

To simulate the fluid flow, the atmosphere is approximated as an incompressible viscous fluid, thus denoting the velocity vector as **v**, the pressure as $p$ and the density as $\rho$ the full Navier-Stokes equations can be reduced to a form like:

$$\begin{aligned} \frac{D\mathbf{v}}{Dt} &= -\frac{1}{\rho}grad(p) + \nu\Delta\mathbf{v} \\ div(\mathbf{v}) &= 0 \end{aligned} \qquad (1)$$

The second equation above is the "continuity equation", expressing conservation of mass, while the term $D\mathbf{v}/Dt$ is a material derivative operator.

To solve equation (1), the atmosphere surrounding the wildfire is discretized using a cartesian spatial mesh adapted to the ground surface and extending several hundreds of meters on the sides and above the fire surface grid (Fig. 8).
The spatial discretization is an odd multiple of the fire discretization, allowing each atmosphere grid point to "cluster" several fire grid points.
Physical quantities are defined at each grid point in terms of wind speed, air temperature, pressure and density. All these quantities are initialized at the first step using the environmental conditions set from IOS (Instructor Operating Station) outside of the domain. Dirichlet boundary conditions assure continuity of the physical variables during the simulation as the aircraft crosses the domain boundaries.

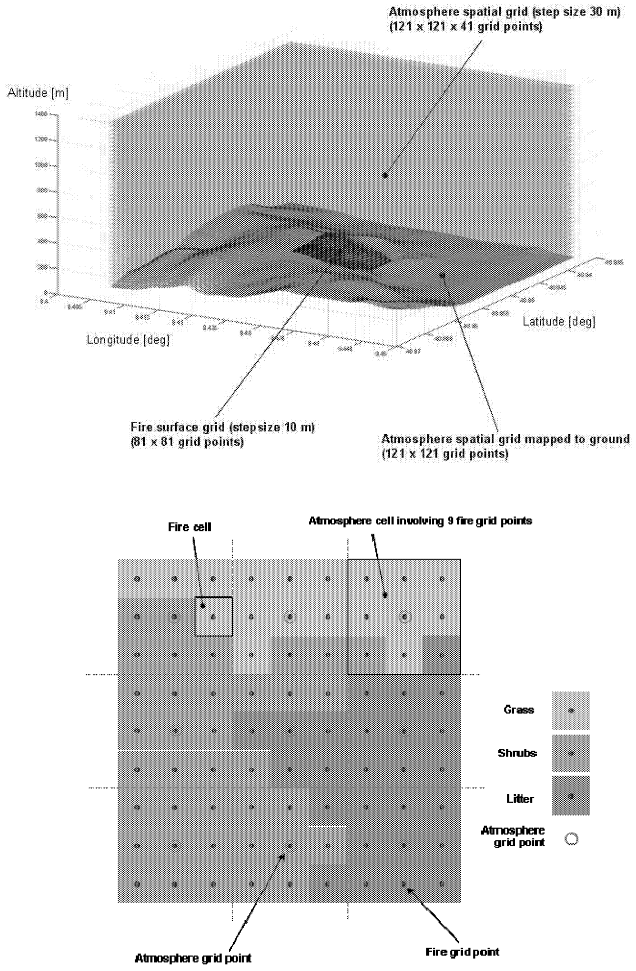

Atmosphere spatial grid (step size 30 m)
(121 x 121 x 41 grid points)
Altitude [m]
Latitude [deg]
Longitude [deg]
Fire surface grid (stepsize 10 m)
(81 x 81 grid points)
Atmosphere spatial grid mapped to ground
(121 x 121 grid points)
Fire cell
Atmosphere cell involving 9 fire grid points
Grass
Shrubs
Litter
Atmosphere grid point
Atmosphere grid point
Fire grid point

Figure 8: Atmosphere and fire grids.

The dynamic evolution of the physical quantities is calculated by successively applying operators reproducing the phenomena represented by the different terms in equation (1) which are:

- Viscosity and pressure effects
- Advection of state variables by the fluid flow
- Thermal diffusion
- Thermal buoyancy

Each operator is defined as follows.

*Viscosity and pressure effects.*
Viscosity causes velocity diffusion while pressure gradients drive the velocity field. Instead of introducing explicitly the pressure as a state variable in the calculations, we followed the approach of Miyazaki (Miyazaki et al. 2001) introducing the discrete version of *grad(div(***v***))*.
Following the notation of Miyazaki, let's denote as $v_x$ the x-component of velocity at the current time step at the grid point with indices i,j,k and $v_x^*$ the same component updated by the viscosity and pressure operator (the y and z components are calculated in the same way with permutations of the indices):

$$v_x^*(i,j,k) = v_x(i,j,k) + k_v \Delta v_x(i,j,k) + k_p grad(div\mathbf{v})_x \qquad (2)$$

Where $k_v$ is the viscosity ration and $k_p$ is the coefficient of the pressure effect. The second and third terms above are:

$$\begin{aligned}\Delta v_x(i,j,k) = \frac{1}{6}[&v_x(i+1,j,k) + v_x(i-1,j,k) + \\ &v_x(i,j+1,k) + v_x(i,j-1,k) + \\ &v_x(i,j,k+1) + v_x(i,j,k-1) - 6v_x(i,j,k)]\end{aligned} \qquad (3)$$

$$\begin{aligned}grad(div\mathbf{v})_x = &\frac{1}{2}[v_x(i+1,j,k) - 2v_x(i,j,k) + v_x(i-1,j,k)] + \\ &\frac{1}{4}[v_y(i+1,j+1,k) + v_y(i-1,j-1,k) \\ &-v_y(i-1,j+1,k) - v_y(i+1,j-1,k) \\ &+v_z(i+1,j,k+1) + v_z(i-1,j,k-1) \\ &-v_z(i-1,j,k+1) - v_z(i+1,j,k-1)].\end{aligned} \qquad (4)$$

*Advection by the fluid flow.*
Advection is incorporated as originally proposed by Stam (Stam 1999) and successively by Miyazaki (Miyazaki et al. 2001) by back-propagating the state variables with the current values of velocities at grid point *i,j,k* and interpolating over the whole 3-D mesh to get the advected value. Even if this method is reported to be unconditionally stable we found that stability problems can occur due to the strong gradients in proximity of the ground if large time steps are adopted. Stability problems have been addressed in a very straightforward way by limiting the maximum wind speed to values consistent with the spatial and temporal discretization.

*Thermal diffusion.*
Thermal diffusion is accounted for by an operator identical to equation (3) above applied to the temperature field with a constant $k_d$.

*Buoyancy.*
Buoyancy is accounted for by adding a vertical velocity at each grid point proportional to the difference between the actual value of temperature at the grid point and a "reference" value (assumed to be the undisturbed atmosphere temperature at that altitude). The coefficient of proportionality is made itself a linear function of altitude

going to zero at a prescribed altitude; this forces the plume shape to be nearly horizontal at the desired altitude and allows a more robust simulation.

This simplification doesn't practically impinge the realism of the simulation, as it is felt by the pilot, since firefighting operations usually happens at low altitude and in the vicinity of the fireline, where strongest gradients exist.

*Other simplifications.*

Several simplifications have been adopted in order to reduce the complexity of the problem and speed up calculations:

- Grid stretching in the vertical direction is not accounted for; this assumption is valid as long as the ground surface is sufficiently "flat" with respect to the grid size.
- The vertical profile of pressure is assumed to remain fixed during the simulation (say an ISA profile); density variations are consequently only due to changes in temperature.
- Wind is initialized with a MIL-F-8785C profile depending only on height above terrain and not taking into account the local orography.

As a fire grows on the ground, the heat released by the burring cells is transferred to the atmosphere grid points by collecting the contributes of all the cells in a cluster. Heat release causes a rise of temperature at the atmosphere grid point, forcing convection through buoyancy and mass conservation but causing also diffusion and advection to the adjacent grid points.

The wind pattern calculated using the CML is fed back to the wildfire model by interpolating wind velocities on the ground; this process closes the interaction between the two models.

## DETAILS OF THE SIMULINK IMPLEMENTATION AND COMPUTATIONAL CONSIDERATIONS

The Fire Simulator basically consists of a Simulink model that implements:

- The fire spread algorithm, written in C code.
- A library, based on BEHAVE system, for predicting spread rate and reaction intensity of wildfires.
- A set of interfaces and optimizations in order to run the simulation in real time mode and to prevent some possible not physical behavior during the simulation execution.

C codes are integrated into the simulink model through the Matlab *Legacy Code Tool*: this method generates a Matlab type dll (*.mexw32) that can be included in the Simulink model.

Fire simulation has to be coupled with Atmosphere simulation to obtain a single Simulink model (.mdl).

Basically, there are two simulation environments: a Simulink stand alone simulation, that runs on a workstation, and the actual Flight Simulator software residing in the host computer, coded in C language. The Matlab Simulink model is ported in C language using the Real Time Workshop Embedded Coder, a Mathworks product that generates compact and fast stand-alone C code.

It is important to underline that the coupled Fire-Atmosphere simulation requires high CPU resources as the grids increases in dimensions, so Multithreading and/or Multitasking techniques should be heavily used during both modeling and code generation phases.

The following figure shows how the two models are coupled in Simulink:

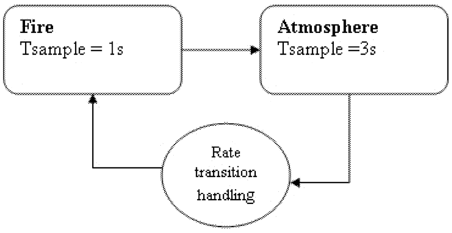


Figure 9: Fire - Atmosphere interaction in Simulink

In order to manage the different sampling rate of the two models, a rate transition block is introduced between them.

In the "General Multithreading Preferences" of Matlab environment, Multitasking option must be enabled.

This setting permits an implicit multithreaded computation that speeds up element wise computations such as those done by the sin and log functions, and computations that use the Basic Linear Algebra Subroutines (BLAS) library, such as matrix multiply. However not all functions in MATLAB are multithreaded, moreover Matlab will never be able to determine if, for example, consecutive function calls in a for-loop are independent of each other.

More complex is to generate code for multithreading, because with the "standard" Real Time Workshop is not possible to generate code optimized for multithread application.

However the multirate model permits to perform some optimizations, starting from the multitasking option that must be enabled on Simulink tasking mode preferences. This setting reflects on the code generation: two main step functions are created (Step0 and Step1).

The final code run on multiprocessor workstations, so the two step functions can be associated with two separate threads (or tasks) running on two separate CPUs.

CPU load on executing Fire model is in the order of 3%. The main algorithm is entirely executed in a scheduler clock cycle (1Hz).

Atmosphere simulation requires a lot of CPU resources, so it is a requisite the all Atmosphere calculations terminate at the next Atmosphere scheduler timing. To ensure that this constraint is satisfied, the Atmosphere task is scheduled at a lower rate (0.33 Hz) than the Fire task.

## VERIFICATION AND VALIDATION

The simulation reported in figure 10 is based on the following input:

- Fire grid: 63 x 63 matrix, each cell with area 100 m$^2$.
- Atmosphere 3D grid: 101 x 101 x 31, stepsize 30 m.

Terrain altitude, slope, aspect, and type are extracted from GIS data and constitute the terrain inputs data for the simulation. The area of interest is a wood in the Maddalena Island, with high density of shrub.

From three ignition cells, fire propagates burning 5.5ha of forest (terrain type: 11 and 12) in about 1h 30min. Fires are ignited under moderate weather conditions: wind was about 3.5 m/s and the terrain characteristics unfavorable for the fire spread (wind blows downhill) leading to a medium fire size in relationship with the elapsed time. By contrast, large catastrophic wildfires are usually driven by severe fire weather conditions with high wind speeds. Note that fire can even propagate to the other side of the road.

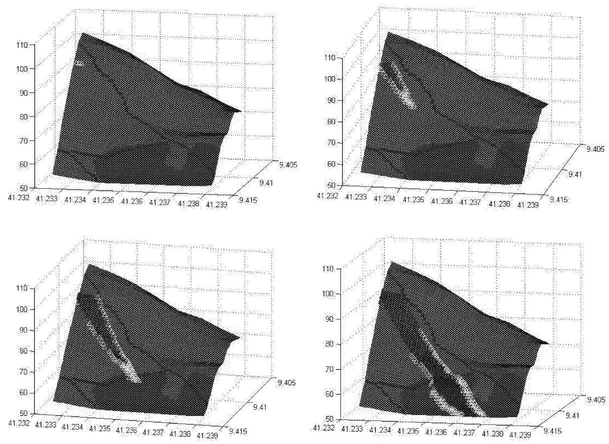


Figure 10: Fire evolution from time 0s to time 5400s.

Simulation results depend on a lot of parameters. Fire shapes simulation on the same terrain can vary significantly simply changing the atmosphere conditions (i.e. wind, moisture), for example winds blowing in direction of rising terrains that hugely favor the fire spread.

A good mean of validation of fire evolution in Maddalena island and neighboring zones, is based on statistically observations of Sardinia fires evolution in the past years. Using the sequence of daily values of fuel moisture and wind, it possible to reproduce fire spread and fire effects with limitations and assumptions of Rothermel's model on the simulation accuracy, described in Andrews (1986).

The following figures show the output of the atmosphere simulation when the fire is completely developed.

The buoyant plume can be clearly identified in Figure 11: the red dot indicates the maximum vertical velocity value of 6.023 m/s. The combined effect of diffusion and advection propagates velocity and temperature downstream of the fire location; the maximum air temperature is 353 K (80 °C) and is located on in the vicinity of the flame core. The flow looks quite viscous since smaller vertical flow structures are absent. Experiments have been conduced introducing Perlin noise or vorticity confinement techniques but this was found to increase significantly the computation time without appreciable effects felt by the pilot. The technique finally adopted to introduce some randomness in the velocity distribution is to keep the atmospheric turbulence active and switch the intensity level according to the velocity gradients encountered by the aircraft.

Figure 12 shows the ground wind pattern in the area of the fire front (about 3.5 m/s wind coming from left in the figure) showing the local wind speed in pseudocolors; the entrainment effect is clearly visible through the convergence of the velocity vectors and the local increase in velocity magnitude.

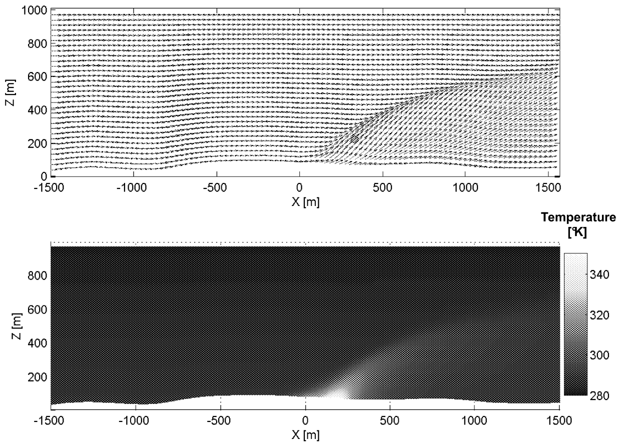


Figure 11: Wind and Temperature distribution in the atmosphere grid.

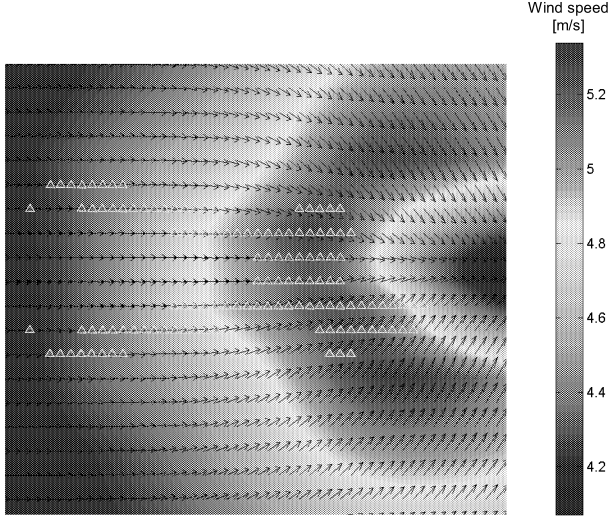


Figure 12: Local wind pattern on ground showing entrainment effect.

## FUTURE WORK

Future improvements of the model would include the implementation of a stretched atmosphere grid to remove the assumption of "flat" terrain; currently a revision of the basic equations is ongoing in order to investigate the possibility of adding grid metric coefficients preserving computational speed. The fire model will be improved adding a model of propagation due to crown fire.

## ACKNOWLEDGEMENTS

This work has been supported by Regione Autonoma Friuli Venezia Giulia, Direzione Centrale Attività Produttive, through a Contribution for Industrial Research (letter n. 9167/2311-28D, project 1848).

## REFERENCES

Albini F., "Estimating Wildfire Behavior and Effects", USDA Forest Service, General Technical Report INT-30", 1976, Intermountain Forest and Range Experiment Station, Forest Station, US Department of Agricolture, Ogden, Utah 84401.

Andrews P.L. 1986, "BEHAVE: Fire Behavior Prediction and Fuel Modelng System – Burn Subsytem", U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station.

Bevins C.D., "fireLib User Manual and Technical Reference".

Coen, J. L., Douglas, C. C., Beezley, J. D., Kremen, R., Mandel, J., Qin, G., Vodacek, A., 2006: Demonstrating the validity of a wildfire DDDAS." *International Conference on Computational Science*.

Coen, J. L. 2005: "Simulation of the Big Elk Fire using coupled atmosphere-fire modeling". *International Journal of Wildland Fire*, 14: 49-59.

Coen, J. L., 2005: "Applications of coupled atmosphere-fire modeling: Prototype demonstration of realtime modeling of fire behavior". *Joint 6th Symposium on Fire and Forest Meteorology/Interior West Fire Council Conference*.

Coen J. L., 2003: "Simulation of wildfire incidents using coupled atmosphere-fire modeling". *5th Symposium on Fire and Forest Meteorology/2nd International Fire Ecology and Fire Management Congress, J2.4*.

Corpo Forestale dello Stato, "Gli Incendi Boschivi 2007", Italy, 2007 (in italian).

Crane K., Llamas I., Tariq S., "Real Time Simulation and Rendering of 3D Fluids," GPU Gems 3, H. Nguyen, ed., chapter 30, pp.633-675, Addison Wesley, Aug. 2007.

Dobashi Y., Kaneda K., Yamashita H., Okita T., and Nishita T., "A Simple, Efficient Method for Realistic Animation of Clouds", *Proc. SIGGRAPH 2000*, 31-37, 2000

Kaneko K., "Simulating Physics with Coupled Map Lattices-Pattern Dynamics, Information Flow, and Thermodynamics of Spatiotemporal Chaos," *Vol. 1, World Scientific,* Singapore, 1990.

McFarland R. E., 1997: "Finite Element Aircraft Simulation of Turbulence", *NASA TM110437*, NASA Ames Research Center, Moffet field, California.

Miyazaki R., Yoshida S., Dobashi Y., and Nishita T., "A Method for Modeling Clouds based on Atmospheric Fluid Dynamics", *Proc. Pacific Graphics 2001*, 363-372, 2001.

Miyazaki R., Dobashi Y., and Nishita T., "Simulation of Cumuliform Clouds Based on Computational Fluid Dynamics", EUROGRAPHICS 2002 , 2002.

MIL-F-8785C, "Flying Qualities of Piloted Airplanes".

Mizuno R., Dobashi Y., and Nishita T. "Volcanic smoke animation using CML". *Proc. of International Computer Symposium 2002*, 2:1375–1382, 2002.

Patton, E. G., Coen, J. L., 2004: "WRF-Fire: A coupled atmosphere-fire module for WRF". *5th WRF/14th MM5 Users' Workshop*, 221-223.

Rothermel, R.C. 1972. "A mathematical model for predicting fire spread in wildland fuels". Res. Pap. INT-115. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station.

Stam J., "Stable Fluids", *Proc. SIGGRAPH'99*, 1999, pp. 121-128.Wu, E., Y. Liu, and X. Liu. 2004. "An Improved Study of Real-Time Fluid Simulation on GPU." In *Computer Animation and Virtual Worlds* 15(3–4), pp. 139–146.

## BIOGRAPHY

**LUCA CISTRIANI** was born in Frascati (Rome), Italy in 1972, and received a Batchelor Degree in Aeronautical Engineering from the University of Rome in 1999. In 2000 started working at Meteor C.A.E. (now joined in Selex-Galileo) in Ronchi dei Legionari (Gorizia), as a UAV design engineer in the Aerodynamics and Flight Mechanics Department. From 2000 to 2007 has been employed in different roles, including Project Leader for the development of the air vehicle and responsible for the flight trials of theFalco UAV System and project leader for the Locusta unmanned target system. From 2003 to 2007 has been in charge of the Flight panel in the certification team of the Falco System with the Italian Civil Aviation authority (ENAC). Currently is in charge of the Aeronautical modeling department in the Simulators Business Area.

E-mail: luca.cistriani@selexgalileo.com

**SEBASTIANO BONFIGLIO** was born in Erice (Trapani), Italy in 1977 and received a Bachelor Degree in Electronic Engineering from the University of Palermo in 2003. In 2005 received a Master of Science in "Advanced Communications and Navigation Satellite Systems" from the University of Tor Vergata in Rome. In 2005 started working at Altran as consultant for Selex Galileo in Ronchi dei Legionari site. Actually he works as system engineer in the Aeronautical modeling department. His work is focused on the deployment of real time simulators systems.

E-mail: sebastiano.bonfiglio@altran.it

**NICOLA STELLA** was born in Matera , Italy in 1979 and received a Bachelor Degree in (Theoretical) Physics from University of Bari in 2008. Specialized on Computational physics and numerical computing in September 2008 started working as free-lance mathematical and numerical consultant. In May 2009 started working at Geotec S.r.l. as Data Analyst, focusing his attention on the application of standard algorithms for remote sensing image processing and development of new algorithm to process SAR and LIDAR Data.

E-mail: nicolastella1@gmail.com

# AUTHOR LISTING

# AUTHOR LISTING

# AUTHOR LISTING