SIMUL 2013

The Fifth International Conference on Advances in System Simulation

ISBN: 978-1-61208-308-7

October 27 - November 1, 2013

Venice, Italy

SIMUL 2013 Editors

Marek Bauer, Cracow University of Technology, Poland

Pascal Lorenz, University on Haute Alsace, France

SIMUL 2013

Forward

The Fifth International Conference on Advances in System Simulation (SIMUL 2013), held on October 27 - November 1, 2013 - Venice, Italy, continued a series of events focusing on advances in simulation techniques and systems providing new simulation capabilities.

While different simulation events are already scheduled for years, SIMUL 2013 identified specific needs for ontology of models, mechanisms, and methodologies in order to make easy an appropriate tool selection. With the advent of Web Services and WEB 3.0 social simulation and human-in simulations bring new challenging situations along with more classical process simulations and distributed and parallel simulations. An update on the simulation tool considering these new simulation flavors was aimed at, too.

The conference provided a forum where researchers were able to present recent research results and new research problems and directions related to them. The conference sought contributions to stress-out large challenges in scale system simulation and advanced mechanisms and methodologies to deal with them. The accepted papers covered topics on social simulation, transport simulation, simulation tools and platforms, simulation methodologies and models, and distributed simulation.

We welcomed technical papers presenting research and practical results, position papers addressing the pros and cons of specific proposals, such as those being discussed in the standard forums or in industry consortiums, survey papers addressing the key problems and solutions on any of the above topics, short papers on work in progress, and panel proposals.

We take here the opportunity to warmly thank all the members of the SIMUL 2013 technical program committee as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and efforts to contribute to the SIMUL 2013. We truly believe that thanks to all these efforts, the final conference program consists of top quality contributions.

This event could also not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the SIMUL 2013 organizing committee for their help in handling the logistics and for their work that is making this professional meeting a success. We gratefully appreciate to the technical program committee co-chairs that contributed to identify the appropriate groups to submit contributions.

We hope the SIMUL 2013 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in simulation research. We also hope the attendees enjoyed the charm of Venice.

SIMUL 2013 Chairs

SIMUL Advisory Chairs

Edward Williams, PMC-Dearborn, USA
Paul Fishwick, University of Florida-Gainesville, USA
Christoph Reinhart, Harvard University - Cambridge, USA
Amr Arisha, College of Business, DIT, Ireland

SIMUL 2013 Research Liaison Chairs

Tae-Eog Lee, KAIST, Korea
Marko Jaakola, VTT Technical Research Centre of Finland, Finland

SIMUL 2013 Industry Liaison Chairs

Diglio A. Simoni, RTI International – RTP, USA
Shengnan Wu, American Airlines, USA
Ann Dunkin, Palo Alto Unified School District, USA
Tejas R. Gandhi, Virtua Health-Marlton, USA

SIMUL 2013 Special Area Chairs

Model-based system prediction
Georgiy Bobashev, RTI International -Research Triangle Park, USA
Aida Omerovic, SINTEF & University of Oslo, Norway

Process simulation
Ian Flood, University of Florida, USA
Gregor Papa, Jozef Stefan Institute - Ljubljana, Slovenia

SIMUL 2013 Publicity Chairs

Nuno Melao, Catholic University of Portugal - Viseu, Portugal

SIMUL 2013

Committee

SIMUL Advisory Chairs

Edward Williams, PMC-Dearborn, USA
Paul Fishwick, University of Florida-Gainesville, USA
Christoph Reinhart, Harvard University - Cambridge, USA
Amr Arisha, College of Business, DIT, Ireland

SIMUL 2013 Research Liaison Chairs

Tae-Eog Lee, KAIST, Korea
Marko Jaakola, VTT Technical Research Centre of Finland, Finland

SIMUL 2013 Industry Liaison Chairs

Diglio A. Simoni, RTI International – RTP, USA
Shengnan Wu, American Airlines, USA
Ann Dunkin, Palo Alto Unified School District, USA
Tejas R. Gandhi, Virtua Health-Marlton, USA

SIMUL 2013 Special Area Chairs

Model-based system prediction
Georgiy Bobashev, RTI International -Research Triangle Park, USA
Aida Omerovic, SINTEF & University of Oslo, Norway

Process simulation
Ian Flood, University of Florida, USA
Gregor Papa, Jozef Stefan Institute - Ljubljana, Slovenia

SIMUL 2013 Publicity Chairs

Nuno Melao, Catholic University of Portugal - Viseu, Portugal

SIMUL 2013 Technical Program Committee

Erika Abraham, RWTH Aachen University, Germany
Chrissanthi Angeli, Technological Institute of Piraeus - Athens, Greece
Amr Arisha, College of Business - DIT, Ireland

Bratislava, Slovakia
Jennie Lioris, CERMICS, France
Francesco Longo, University of Calabria, Italy
Jose Machado, Universidade do Minho, Portugal
Ricardo Marcelín-Jiménez, Universidad Autónoma Metropolitana, Mexico
João Pedro Jorge Marques, University of Porto, Portugal
Goreti Marreiros, Engineering Institute - Polytechnic of Porto, Portugal
Stefano Marrone, Seconda Università di Napoli, Italy
Marco Massetti, Universitat Ramon Llull - Barcelona, Spain,
Don McNickle, University of Canterbury - Christchurch, New Zealand
Nuno Melao, Catholic University of Portugal - Viseu, Portugal
Jürgen Melzner, Bauhaus-University Weimar, Germany
Marco Mevius, HTWG Konstanz, Germany
Lars Mönch, University of Hagen, Germany
Muaz Niazi, Bahria University, Pakistan
Libero Nigro, Università della Calabria, Italy
Mara Nikolaidou, Harokopio University of Athens, Greece
Michael North, Argonne National Laboratory, USA
Aida Omerovic, SINTEF ICT, Norway
Gregor Papa, Jozef Stefan Institute - Ljubljana, Slovenia
Laurent Perochon, VetaGro Sup, France
Claudine Picaronny, LSV ENS Cachan, France
Henri Pierreval, IFMA-LIMOS, France
François Pinet, Irstea, France
Marta Pla-Castells, Universitat de València, Spain
Katalin Popovici, MathWorks Inc., USA
Francesco Quaglia, Sapienza Universita' di Roma, Italy
Urvashi Rathod, Symbiosis Centre for Information Technology, India
Cláudia Ribeiro, INESC-ID Lisbon, Portugal
José Luis Risco Martín, Universidad Complutense de Madrid, Spain
Agostinho Rosa, Technical University of Lisbon, Portugal
Rosaldo J. F. Rossetti, University of Porto, Portugal
Manuel Filipe Santos, University of Minho, Portugal
Jean-Francois Santucci, University of Corsica, France
Guodong Shao, National Institute of Standards and Technology - Gaithersburg, USA
Larisa Shwartz, IBM T. J. Watson Research Center - Hawthorne, USA
Jeffrey S. Smith, Auburn University, USA
Eric Solano, RTI International, USA
Nary Subramanian, University of Texas at Tyler, USA
Elena Tànfani, University of Genova, Italy
Alexander Tatashev, Moscow University of Communications and Informatics, Russia
Pietro Terna, University of Torino, Italy
Michele Tizzoni, Institute for Scientific Interchange, Torino, Italy
Alfonso Urquia, Dept. Informatica y Automatica - UNED, Spain

Copyright Information

For your reference, this is the text governing the copyright release for material published by IARIA.

The copyright release is a transfer of publication rights, which allows IARIA and its partners to drive the dissemination of the published material. This allows IARIA to give articles increased visibility via distribution, inclusion in libraries, and arrangements for submission to indexes.

I, the undersigned, declare that the article is original, and that I represent the authors of this article in the copyright release matters. If this work has been done as work-for-hire, I have obtained all necessary clearances to execute a copyright release. I hereby irrevocably transfer exclusive copyright for this material to IARIA. I give IARIA permission or reproduce the work in any media format such as, but not limited to, print, digital, or electronic. I give IARIA permission to distribute the materials without restriction to any institutions or individuals. I give IARIA permission to submit the work for inclusion in article repositories as IARIA sees fit.

I, the undersigned, declare that to the best of my knowledge, the article is does not contain libelous or otherwise unlawful contents or invading the right of privacy or infringing on a proprietary right.

Following the copyright release, any circulated version of the article must bear the copyright notice and any header and footer information that IARIA applies to the published article.

IARIA grants royalty-free permission to the authors to disseminate the work, under the above provisions, for any academic, commercial, or industrial use. IARIA grants royalty-free permission to any individuals or institutions to make the article available electronically, online, or in print.

IARIA acknowledges that rights to any algorithm, process, procedure, apparatus, or articles of manufacture remain with the authors and their employers.

I, the undersigned, understand that IARIA will not be liable, in contract, tort (including, without limitation, negligence), pre-contract or other representations (other than fraudulent misrepresentations) or otherwise in connection with the publication of my work.

Exception to the above is made for work-for-hire performed while employed by the government. In that case, copyright to the material remains with the said government. The rightful owners (authors and government entity) grant unlimited and unrestricted permission to IARIA, IARIA's contractors, and IARIA's partners to further distribute the work.

Table of Contents

A Standardized Simulation Model with Strategic Approach for Distribution Networks

A Case Study in Mexico

Homero H. Contreras, José Pablo Nuño
Interdisciplinary Graduate Programs and Research Center
UPAEP University
Puebla, Mexico
homerohector.contreras@upaep.edu.mx;
pablo.nuno@upaep.mx

Eric Porras, Eduardo Zelaya
EGADE Business School
ITESM – Campus Santa Fe
Mexico City, Mexico
eric.porras@itesm.mx; ezelaya@itesm.mx

***Abstract*— Considering that analytic tools are not completely suitable to analyze supply chain and distribution networks, simulation is considered a better alternative. Some theories about discrete simulation have been suggested, especially those related to the use of standardized models and the use of strategic planning process in simulation. This paper presents a standardized simulation model based on a simulation programming language rather than a graphical simulator, to be used as a decision-making tool for the top management due to its strategic approach. The model is validated in a real business case, where tangible results were achieved.**

Keywords-Simulation; standardized model, distribution network, strategy

I. INTRODUCTION

Considering the opportunities that distribution network presents to create value and profits to any organization, specific tools to analyze and improve distribution must be used in today's business environment. Most tools might be classified as analytical tools (those using a closed-form solution based on a mathematical algorithm) or simulation tools [1]

Simulation is considered as a suitable tool because the integration of dynamic and stochastic issues of real life processes is a critical task. Standardized simulation models are those which can be applied to a broad range of systems and, at the same time, they can be adjusted to different scenarios and performance criteria, becoming specific when data for a particular system is loaded [2]. Therefore, a model is suggested, based on a common logic used to evaluate the configuration of a distribution network. This evaluation is focused on a strategic planning approach, using a general purpose simulation programming language.

Any model must be evaluated through specific key performance indicators, which should be similar to the intended use case.

This paper is organized as follows: a literature review about simulation, strategy and supply chain is presented, supporting the proposed idea; then, the methodology used to define a standardized model, including the objectives, logic and specific considerations of the code is described. The validation through a real business case in Mexico is included, and finally, some conclusions and future research are presented.

II. LITERATURE REVIEW

Any organization can been considered as a series of related operations where its assets must be adapted to the actual and future demand, in several levels of aggregation and time horizons [3]; therefore, the supply chain management becomes an integral part of the strategy of the organizations. In particular, the distribution network plays a vital role because successful firms have been supported by competitive advantages related to the optimization between demand and delivery [2]. Reaching flexibility in distribution and evaluating the potential scenarios that can be faced in delivery [4] are also relevant issues in the supply chain.

Another important issue in supply chain deals with the integration of suppliers, producers, warehouses and point of sales. This integration also deals with the manufacturing and distribution of goods or services on time, on the exact amount and in the precise place, considering a minimum cost and a suitable service level [5]. Inventories are also a critical issue affecting the supply chain, and become an even more relevant factor in retail industries [6].

Any supply chain is a stochastic, dynamic and complex system facing a high variability and uncertainty, as well as a disperse configuration [7]. Therefore, it is mandatory to consider strategic decisions and specialized tools to support decision making process [3], focusing in either costs or differentiation [8].

The integration of all the activities within a distribution network provides opportunities in creating value, reducing costs, raising productivity and maximizing profits; however, this integration cannot be evaluated using analytic models [10]. Some analytic models can be used with a limited confidence and within specific constraints in the integration of variability [9], but are not very useful.

Gongtao and Gavirneni [11] have suggested a model to evaluate distribution policies based on Erlang distribution; an analytic model based on a recursive approach to analyze demand, inventories and deliveries, but limited to normal distribution, is presented in Kim et al. [12].

 ISBN: 978-1-61208-308-7

Considering that analytic models cannot accurately represent the real and complex behavior of the supply chain, simulation is considered as a suitable technique to analyze and evaluate it. Simulation provides a deep understanding of the system, as well as experimental ways to support a decision which considers variability [3]. It is important to notice that simulation does not provide an optimum result, but it provides the evaluation of several scenarios and how they affect some specific Key Performance Indicators (KPI), as stated by Fleisch and Tellkamp [13]. Also, the definition of specific and relevant KPI's is also a vital issue.

Some authors have suggested that simulation models based on Systems Dynamics (SD) theory can be used to evaluate strategies related to supply chain [4]. However, these models are focused on continuous behavior, with a high level of details and complex relations, making them more difficult to design, analyze and improve. In Labarthe et al. [9], the use of SD and agent-modeling has been suggested, but under an operational approach. Considering that most of the typical operations in a supply chain occur at specific points of time, the discrete simulation provides a better option to analyze it.

Siebert and Zubanov [14] have used discrete simulation to integrate fluctuations in demand throughout days, weeks and months, but using correlations, hence limiting the stochastic behavior of the model.

Zhang and Zhang [5], have proposed a base model, but restricted to three echelons due to the complexity of the integration of more echelons. Almeder and Preusser [15] have presented a simple simulation model, where a lineal-deterministic algorithm is optimized to prepare input data which is returned to the simulation model; however, this is not a simulation tool per-se, but a hybrid model.

Hafeez et al. [16] propose the decomposition of the supply chain in two echelons, mainly to simplify the model, but they do not provide any arguments to support this idea; they also suggested an inventory approach totalizing the inventory levels across the network, similar to multiechelon theory, without any deep analysis on this issue.

Some authors have proposed the use of standardized simulation models. These models, when used in distribution networks, must be flexible, based on parameters, efficient in computing requirements and repetitive so they can change the position within the network, as suggested by Longo and Mirabelli [17]. Standardized models are based on the idea that there are always some common processes within the distribution network that can be reused [7], and they must be focused on the specific elements of the supply chain that will be considered in the analysis [4]. However, they should consider the complete environment of the system, not restricted to a very limited approach [13].

Standardized simulation models present a challenge because most of the actual software available is highly graphics dependent and based on objects. These characteristics make software easy to use and learn, but implementing some logical processes (e.g., loops or complex conditionals) might be difficult; therefore, external applications or programming-languages must be used to fulfill a standardized model [18]. Simulation programming languages provide an easy way to create a detailed logic; Yang [19] even asseverates that actual software is based on the languages developed in the 60's and 70's and sometimes these old languages are even used to process the logic. As a matter of fact, SLAM [20], GPSS [21] and SIMAN [22] are examples of software that are used today and provide useful results. They are robust and they have been taught in universities, but their market-share is very low today, compared to newest software [19].

Considering that standardized models can be easily applied through the use of simulation programming languages, this paper proposes the design of a simulation model applied to a distribution network through the simulation language SIMNET [23]. This model is tested in a business case focused on health industry because this sector provides opportunities in inventories and continuous improvement [6] and it represents an important component of the gross domestic product [24]. It is also a relevant industry in developing countries [25].

III. PROPOSED MODEL

A standardized simulation model, codified in SIMNET II and under a strategic approach, is proposed in this section.

The model is based on a standard logic between a two-echelon structure of the distribution network, providing a base code. This code should be general enough to become specific with real data, and it must be used in a recursive way to develop the structure of the complete distribution network.

The methodology used in the design of this standard model is based on the analysis of the system and the definition of objectives and indicators; then, the standard logic must be outlined, and the specific characteristics of the model should be included in the structure of the programming code. Then, a process to verify and validate the code should be used, followed by the design of experiments to improve the systems. Finally, conclusions about the results obtained must be discussed.

The characteristics of the suggested model are presented in the following sections.

A. Objective of the Model and Indicators

The focus of the model is based on the cost strategy, as stated by Porter [8].

Considering the wide spectrum of decisions facing in SCM, a selection of specific KPI related to cost strategy, must be defined before starting the analysis. Two KPI are proposed:

- The inventory levels and,
- The transportation cost.

The integration of both the inventory levels and transportation costs will be made through a total cost indicator. This indicator will be the sum of the cost of holding inventory, the financial cost and the transportation cost. Some inputs for the model must be processed by analytical means in external tools, and the conjunction of simulation and analytical methods might provide the basis for a future hybrid model, as stated by Shanthikumar and Gargent [26].

 ISBN: 978-1-61208-308-7

B. Standard logic

Considering the ideas of Pundoor and Hermann [7], the model encompasses some common processes found on most of the inventory and replenishment systems in all major SCM systems. These processes constitute the main logic that is considered in a two-echelon situation, where one supplier and "n" clients are found. The main logic is the following:

a) At the start of each week, the clients, accordingly to their desired maximum inventory level and the on-hand inventory of the last week, place an order to the supplier.

b) The supplier, also accordingly to its desired maximum inventory level, places an order in to his system, and receives his order from the previous week.

c) When the supplier receives his order, his initial inventory position is calculated, adding to the actual inventory level the received order.

d) The clients receive their orders from the supplier, according to one of the following conditions:

- If the supplier's initial inventory level is greater than the sum of all orders received, the clients receive the total amount of pieces requested.
- Otherwise, the clients receive a proportional amount of their order, based on the per cent that each order represent of the sum of all orders requested to the supplier.

g) Orders are transferred from the supplier to the clients, and stock base at the supplier is updated.

h) The initial inventory of the week is updated for each client.

i) A demand is generated for each client, and the final inventory of the week is calculated.

j) The average inventory level of the week is calculated for each client.

k) The units sold for each client are calculated, and if needed, emergency orders are placed to the suppliers to fulfill the complete demand. These emergency orders are supplied from the stock base in the supplier, and if the stock base is not enough to fulfill all emergency orders, they are prorated based on a percentage basis (in this last case, the stock base ends at zero).

l) The average inventory level at the supplier is updated.

This two-echelon logic can be replicated into a series of suppliers-clients structure, where a supplier becomes a client of another supplier, as seen in Fig. 1.

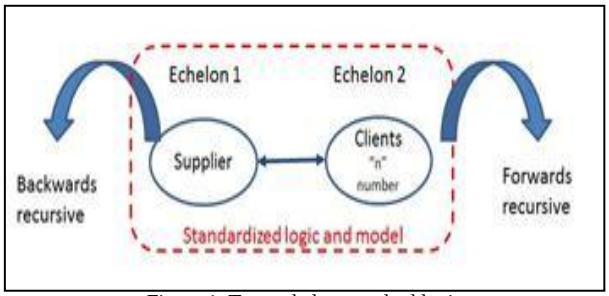


Figure 1. Two-echelon standard logic

However, in order to recreate the structure of a distribution network, the replication should also be used in a parallel framework. For example, a network structure might be based on a Master Distribution Centre (MDC) which serves some Regional Distribution Centres (RDC), and these RDC might serve Regions or Clients. Therefore, the model should start at the end of the network, considering the aggregated demand of the regions, and then move backwards to the MDC in a series of phases, as stated in Fig. 2.

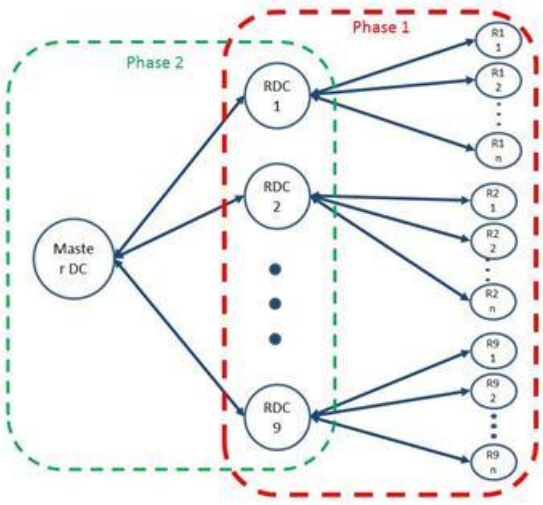


Figure 2. Replication of the standard logic

C. Model characteristics

Some of the opportunities identified in the literature for the usage of simulation in supply chain are included in the model. In particular, the idea to reuse a generic model based on a common logic, as stated by Fordyce et al. [3] and the strategic approach cited by Chang and Makatsoris [27] and by Pundoor and Hermann [7]. The simplification approach to improve the computational performance is also include, as suggested by Longo and Mirabelli [17]. Other characteristics of the model are cited below.

1) Aggregation and strategic approach

Considering the design of a distribution network as a strategic issue, some assumptions must be made. In particular, some operational issues and issues that affect the long term results must be included. For example, all the pieces and products to be demanded must be added into a single aggregated demand, without distinguishing between individual items. This approach allows to be focused in the total items hold in inventory throughout the whole system, and also in the performance of the inventory levels of the complete network.

2) Unitary transportation cost

Distribution network might use different types of vehicles and routes. Therefore a unitary transportation cost for each RDC should be defined.

3) Discrete operation

Because most operations considered in any supply chain occur at specific points of the time, a discrete approach is

used. According to this, all variables are considered as observation based ones, even those typically considered as time based variables (computational rules have been defined to provide this conversion).

Strategic issues also provides a justification to this conversion because the detailed behavior of the inventory level is not required, due to the aggregate approach of the analysis.

The model is based on a single control entity that flows through the code and executes each of the logic steps previously defined.

All the previous characteristics also supports the foundation of a simulation model that is completely integrated and do not require connection to external data. Another benefit of this model is the fast execution because memory allocation of observation based variables is considerable lower than time based ones.

4) Time framework

Considering that the focus of the analysis is the entire cycle of operations based on a discrete time scope, the model uses a non-temporal time framework. Therefore, the case to be analyzed might be based on a week, a month or a day, depending on the desired simulation analysis.

5) Simulation software

The standard logic must deal with several conditionals, mathematical operations and loops between some relations supplier-clients. Therefore a straightforward way to carry out the model is required, in both serial and parallel operations. Most graphical simulators available (e.g., Arena, ProcessModel, Simio) provide an easy-of-use environment, but cannot deal with loops and conditionals in a simple way.

A simulation programming language is not as friendly as a graphical simulator, but it provides several important advantages:

- Additional flexibility,
- Allows a self-contained code,
- All logic, operations, calculations and data exchange can work in an integrated way within the model,
- Easy debugging and,
- The model can run faster.

The proposed model is developed using SIMNET II, a simulation programming language owed to Dr. Hamdy Taha. SIMNET II is based on the use of four special nodes linked by branches. The nodes used in SIMNET II are:

- A source where entities arriving into the model are created,
- A queue which serves to house waiting entities,
- A facility, where service is performed and,
- An auxiliary, which is a special node with infinite capacity providing additional flexibility to the simulation.

Branches connect the nodes and control the flow of entities. During the running of the model, branches can perform logic and several operations, including special routing of entities, evaluation of conditionals, execution of loops (for—next and do-while), evaluation and assignments of control variables, collection of statistical variables or exporting/importing data.

A specific characteristic of SIMNET II is the use of the so-called PROCEDURES. These structures deal with the modeling of repetitive segments in a compact and efficient way, and can be considered as the foundations of a standard model, considering a logic that can be automatically replicated both in series or parallel, thus adding flexibility in reusing code. Through the use of procedures, the complete distribution network can be analyzed in small parts or in the whole, starting at the downstream of the supply chain and continuing upstream, just defining the appropriate input data.

IV. VALIDATION OF THE MODEL

In order to test the standardized model, a business case is used involving a Mexican company which runs a network of about 2000 drugstores in the whole country (about 25% own and 75% franchisees), and under the specific conditions of this firm.

The distribution network is based on one MDC serving nine RDC, each one serving specific geographical areas of the country (mainly complete States). Each State, and its complete number of drugstores included, is considered a "region".

This firm has its own-laboratory which produces most of the drugs and products to be sold. This laboratory is located next to the MDC, the only facility to which it serves, and there is no distribution cost to the MDC. The rest of the drugs and products are supplied by independent and external laboratories, which directly deliver to each of the RDC. The transportation cost of these products is carried out by the suppliers and included in the unitary cost to be paid by the company.

Each RDC directly serves and deliver products to own-stores, local warehouses and big franchisees. Due to the reduced amount of purchases, small and medium size franchisees are treated as final customers.

A graphical representation of the distribution network of this company is presented in Fig. 3:

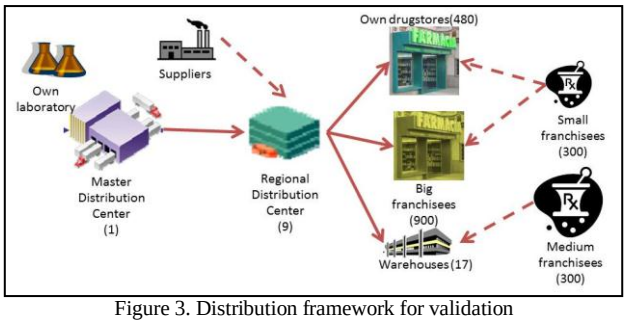


Figure 3. Distribution framework for validation

A special case is presented by small and medium size franchisees. As long as they are treated as customers, they must pick-up merchandise at specific drugstores or warehouses. Therefore they do not affect the distribution cost of the total network.

The distribution is made through an external delivery company, serving the whole network by trucks, and some

areas by ferry. The routes are defined by each RDC and the process of putting orders and delivering products is made only once per week, with delivery time being less than three days (with average of two days in the complete network). A specific methodology to define a unitary transportation cost for each RDC is defined, regardless of the vehicle and route.

A periodic review system is used for the inventory control, based on a period of one week. The order-up-to-level approach is also employed, and the maximum level of inventory desired is based on a heuristically philosophy of 30 days of sales. Of course, this system produces a high level of inventory, but also provides a fill rate of 99.9%

The model is run to resemble the actual distribution network, under a steady state analysis with a 95% of confidence interval.

Considering the two KPI defined in section 2, the results of the model versus the historical data are within the 5% error-tolerance. Because numerical data of the company are confidential, comparisons are made using percentages, where 100% represents the historical data.

The inventory levels of each RDC and the complete network can be seen in Fig. 4, where an overall difference of about 2.7% can be found.

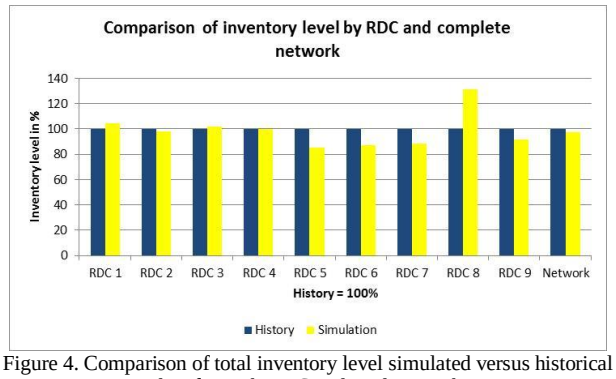


Figure 4. Comparison of total inventory level simulated versus historical data for each RDC and total network

There is one significant difference for RDC number eight in the previous figure. This is caused because this specific RDC was recently open and no enough data are available.

The distribution cost comparison is presented in Fig. 5, where the overall difference is about 3.2%.

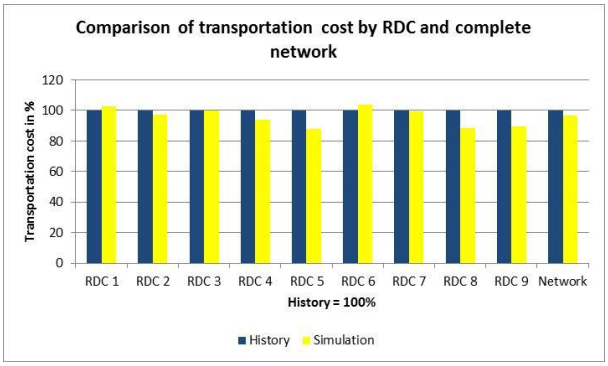


Figure 5. Comparison of total distribution cost simulated versus historical data for each RDC and total network

The analysis has demonstrated that the simulation model is valid and it resembles the actual behavior of the real system within the desired tolerances.

V. CONCLUSIONS AND FUTURE WORK

Considering that the maximum error found on the total inventory level is 2.7% versus the historical data, and deviation in the total transportation cost is only 3.2%, the standardized simulation model works properly and presents a reasonable precision on the assumed confidence interval.

The computational power needed to compile this model is minimum, due to the use of a simulation language versus a graphical simulator.

It has been demonstrated that the standard logic defined in this paper can be used as a base case for distribution network under similar conditions, and might be adjusted with minimum changes to other scenarios. Furthermore, the flexibility provided by the simulation language SIMNET II and its procedures have provided the fundamental bases for a standardized model.

A future research or application of the model might be its use as a decision tool to support the reconfiguration of the actual distribution network. Strategic issues, like evaluation of effects due to open, close and/or merge RDC, or reassignment of regions to each RDC, can be easily carried out in the simulation model, and multiple scenarios could provide a framework to define a new network.

ACKNOWLEDGMENT

CONACYT has granted a Ph.D. scholarship to one of the authors.

REFERENCES

[1] A. Law and D. Kelton, Simulation Modeling and Analysis, Boston, USA: McGraw Hill, 2000.

[2] D. Cope, M. Sam-Fayez, M. Mollaghasemi, and A. Kaylani, Supply chain simulation modeling made easy: An innovative approach, Proceedings of the 2007 Winter Simulation Conference, Dec. 2007, pp. 1887-1896.

[3] K. Fordyce, A. Degbotse, J. Milne, R. Orzell, and C. Wang, The ongoing challenge - An accurate assessment of supply linked to demand to create an enterprise-wide end to end detailed central supply chain plan, Proceedings of the 2008 Winter Simulation Conference, Dec. 2008, pp. 2267-2270.

[4] Q. Wang and N. Ingham, A discrete event modelling approach for supply chain simulation. International Journal of Simulation Modeling, vol. 7, Sep. 2008, pp. 124-134.

[5] C. Zhang and C. Zhang, Design and simulation of demand information sharing in a supply chain, Simulation Modelling Practice and Theory, vol. 15, Jan. 2007, pp. 32-46.

[6] M. Villette, P. Khadgi, R. Moraga, E. Asoudegi, and O. Ghrayeb, Simulation in retail: A case study for process improvement in the receiving area, Proceedings of the 2009 Winter Simulation Conference, Dec. 2009, pp. 2920-2930.

[7] G. Pundoor and J.W. Herrmann, A hierarchical approach to supply chain simulation modeling using the supply chain operations reference model, International Journal of

 ISBN: 978-1-61208-308-7

Simulation and Process Modelling, vol. 2, Jul. 2006, pp. 124-132.

[8] M. Porter, What is strategy?, Harvard Business Review, vol. 74, June 1996, pp. 61-78.

[9] O. Labarthe, B. Espinasse, A. Ferrarini, and B. Montreuil, Toward a methodological framework for agent-based modelling and simulation of supply chains in a mass customization context, Simulation Modelling Practice and Theory, vol. 15, Feb. 2007, pp. 113-136.

[10] F.T.S Chan. and H.K. Chan, The future trend on system-wide modelling in supply chain studies, International Journal of Advanced manufacturing Technology, vol. 25, Feb. 2005, pp. 820-832.

[11] L. Gongtao and S. Gavirneni, Using scheduled ordering to improve the performance of distribution supply chains, Management Science, vol. 56, Sep. 2010, pp. 1615-1632.

[12] G. Kim, D. Chatfield, T. Harrison, and J. Hayya, Quantifying the bullwhip effect in a supply chain with stochastic lead time, European Jornal of Operational Research, vol. 173, Sep. 2006, pp. 617-636.

[13] E. Fleisch and C. Tellkamp, Inventory inaccuracy and supply chain performance: A simulation study of a retail supply chain, International Journal of Production Economics, vol. 95, Mar. 2005, 373-385.

[14] W. Siebert and N. Zubanov, (2010). Management economics in a large retail company. Management Science, vol. 56, Aug. 2010, pp. 1398-1414.

[15] C. Almeder and M. Preusser, A toolbox for simulation-based optimization of supply chains, Proceedings of the 2007 Winter Simulation Conference, Dec. 2007, pp.1932-1939.

[16] K. Hafeez, M. Griffiths, J. Griffiths, and M. M. Naim, Systems design of a two-echelon steel industry supply chain, International Journal of Production Economics, vol. 45, Aug. 1996, pp. 121-130.

[17] F. Longo and G. Mirabelli, An advanced supply chain management tool based on modeling and simulation, Computers & Industrial Engineering, vol. 54, Apr. 2008, pp. 570-588.

[18] C. Jenkins and S. Rice, Resource modeling in discrete-event simulation environments: A fifty-year perspective, Proceedings of the 2009 Winter Simulation Conference, Dec. 2009, pp.755-766.

[19] M. Yang, Using data driven simulation to build inventory model, Proceedings of the 2008 Winter Simulation Conference, Dec. 2008, pp. 2595-2599.

[20] A. Pritsker, Introduction to Simulation and SLAM II, New Jersey, USA: John Wiley & Sons, 1995.

[21] T. Schriber, Simulation using GPSS, New Jersey, USA: John Wiley & Sons, 1974.

[22] D. Pegden, R. Sadowski and R. Shannon, Introduction to Simulation using SIMAN, Boston, USA: McGraw Hill, 1995.

[23] H. Taha, Simulation with SIMNET II, SimTech, Inc., USA. 1992.

[24] J. Swain, Software survey: Simulation - back to the future, ORSM-Today. Vol. 15, May 2011, pp.56-69.

[25] C.K. Prahalad, Serving the world's poor, profitably, Harvard Business Review, vol. 80, Sep. 2002, pp. 48-57.

[26] J. Shanthikumar and R. Gargent, A unifying view of hybrid simulation/analitic models and modeling, Operations Research, vol. 31, Jun. 1983, pp. 1030-1052

[27] Y. Chang and H. Makatsoris, Supply chain modelling using simulation, International Journal of Simulation Systems, Science and Technology, vol. 2, Jan. 2001, pp. 24-30.

 ISBN: 978-1-61208-308-7

Hycon 2 Network Show Case: Sugar Factory

A. Rodriguez, L. F. Acebes, R. Mazaeda, A. Merino, and C. de Prada
Systems Engineering and Automatic Control Department
University of Valladolid
Valladolid, Spain
alexander.rodriguez@autom.uva.es; felipe@autom.uva.es; rogelio@cta.uva.es; alejandro@cta.uva.es; prada@autom.uva.es

***Abstract*— The paper describes a dynamic simulator of a scaled sugar factory to use as a reference, or benchmark, to design and test complex systems control strategies. The process is a subset of a general sugar production process that contains both continuous and batch process units and is closely linked to the factory energy consumption and the quality of the produced sugar. The control problem and the indexes to measure the process performance are set up and the simulation scenarios, or study cases, are given. Finally, different ways to use the simulator from the software point of view are outlined.**

Keywords-process-industry; benchmarking; hybrid systems control

I. INTRODUCTION

HYCON2 Network of excellence [1], supported by the European Union Seventh Framework Programme, has as aims stimulating and establishing the long-term integration of the European research community, leading institutions and industry in the strategic field of control of complex, large-scale, and networked dynamical systems. It has identified several applications domains: transportation, energy, and biological and medical systems.

HYCON2 organizes in ten Work Packages (WP), the WP V is related to benchmarking for testing and evaluating the technologies developed in the network. In particular, two show-case applications corresponding to real-world problems have been selected: the freeway network around the Grenoble area and the high-level operation of two coupled sections of a sugar factory.

In the field of process control, different benchmarks can be found. Some of them are used to controller design [2][3]; others are process-identification oriented (for instance de Wiener-Hammerstein benchmark); control plant-wide is another subject for some of them [4].

In our proposal, a system oriented to design high level controllers, which includes plant scheduling, operation and economic optimization, has been thought. The sugar factory show case considers significant problems and elements of a real sugar plant combining sections with continuous and batch process units. The low level regulatory control system is given and the researchers must concentrate on designing methods and algorithms to operate the plant optimally, according to a set of economic targets and observing a set of constraints. Four performance indexes, to compare the solutions, are given. They measure the energy cost, the economic profit and the productive capacity of the system.

The simulation model has been implemented in EcosimPro [5][6], that is an Object Oriented Modeling and Simulation Tool with similar characteristics of any language that implements Modelica [7][8][9], reusing model components from sugar process model libraries [10][11][12]. The simulation model cannot be validated versus real data, because the real system doesn't exist. However, the library of models has been used to develop a simulator to train plant operators of sugar factories. This training simulator has been partially quantitative validated with real data and totally qualitative validated by managers and control plant operators [13][14][15].

The simulator is available from EcosimPro or MATLAB-SIMULINK environment and, additionally, a standalone version with a Supervisory Control and Data Acquisition (SCADA) interface is available too. The standalone version can communicate with any software tool that works as an OLE for Process Control (OPC) server [16].

The paper is organized as follows: in Section 2, a description of the simulated process and the control problem is outlined; Section 3 explains the show case performance indexes and study cases; Section 4 deals with the computer subjects and, finally, some conclusions are given.

II. PROCESS DESCRIPTION

A. Sugar Production Process

A typical beet sugar factory (Fig. 1) is divided in two great sections: the Beet End and the Sugar End [17][18]. The Beet End contains the diffusion, purification and evaporation stages. The sucrose is extracted from beets by using a diffusion process, the removal of as many impurities as possible is carried out in the purification section and the concentration of the resulting sucrose solution is achieved in the evaporation section, which uses live steam. In the Sugar End is where the crystallization of the dissolved sucrose is carried out to deliver the white sugar grains with commercial value. The crystallization is performed in batch and continuous crystallizers that are heavy consumers of the steam, which is served by the evaporation. Other parts of a standard sugar factory are the boilers and turbo generators, where fuel or gas is used to obtain electric energy and live steam, and the pulp dryer, where the exhausted beets are dried to obtain food for the cattle.

For the show case, only a subset of the real process has been selected: the evaporation and crystallization sections. Although the proposed process is smaller than a real factory, the control problem is very significant, because of the great interaction between sections, the continuous mode operation of the evaporation section and the semi-batch operation in the crystallization one.

 ISBN: 978-1-61208-308-7

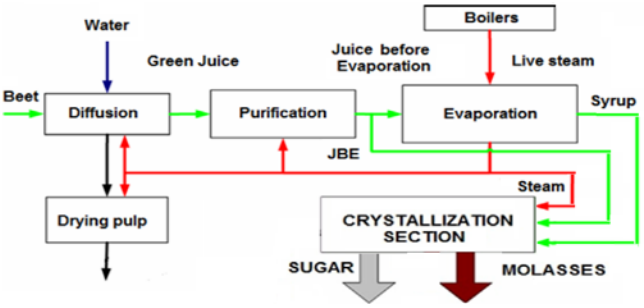


Figure 1. Schematic of a sugar factory

B. *Evaporation Section*

The aim of this section is to increase the Brix (matter content in sugar solution) of the juice (sugar solution), by evaporating the water to obtain syrup. The required heat is provided by a flow of fresh steam that comes from the boilers after passing by electricity generating turbines. Usually, the evaporators are grouped in four or more effects. An effect is a cascade of evaporators, in which the juice flows from one to other one, but using the same source of steam.

In the show case, each effect contains only one evaporator and it's a three effect arrangement (Fig. 2), in which the juice circulates in series increasing its Brix up to a certain value. The first evaporator receives its heating steam from the boilers, but the heating of the other evaporators is provided by the steam produced in the previous one. The scheme is energy efficient in the sense that allows the multiple reuse of the live steam that comes from the factory boilers. It is important to mention that part of the steam produced in each evaporator is redistributed in the factory to fulfill other technological duties. In particular, the most important consumers of the steam generated by the evaporation section (steam I and II) are the crystallizers of the Sugar End.

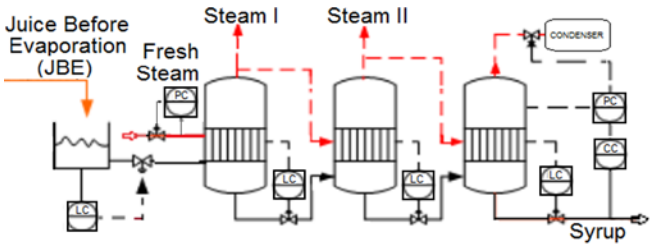


Figure 2. The show case Evaporation Section

With respect to the low level control structure, the Brix of the syrup at the output is controlled by a cascade arrangement of two Proportional Integral and Derivative (PID) controllers; the outer loop (CC) gives the reference of the inner loop (PC), which controls the vacuum pressure in the last effect chamber. Besides, the juice level in the evaporators and in the feeding tank is maintained by level controllers (LC). Additionally, there is a controller to set the value of the fresh steam pressure (PC).

C. *Crystallization Section*

In modern factories, the Sugar End, usually, has an architecture consisting of three stages: the first, or A stage, is dedicated to the production of commercial white sugar crystals and the rest to the exhaustion of the remaining syrup.

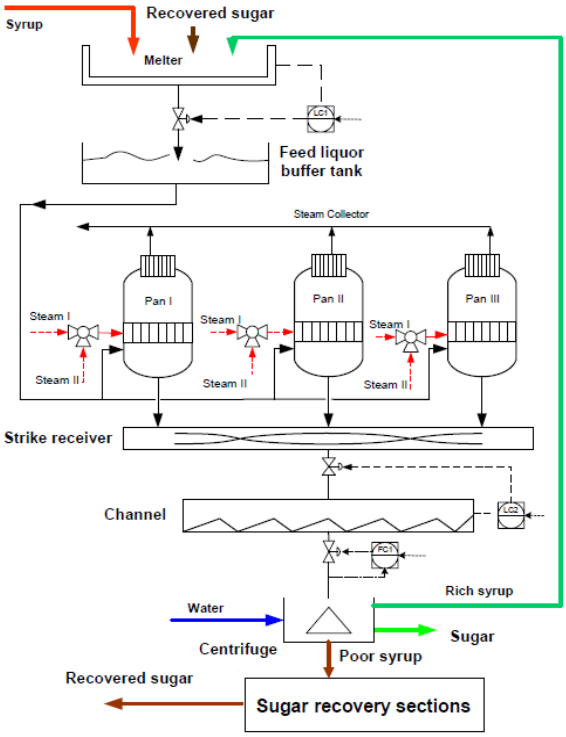


Figure 3. The show case Sugar End (Crystallization Section)

The show case, exclusively, considers A stage (Fig. 3). The syrup from the evaporation stage is sent to the melter plus a flow of recycled rich syrup and another one of low-quality sugar from the B and C stages, with the purpose of obtaining the so called standard liquor. The latter is discharged in a storage tank, which serves the feed syrup to the crystallizers or pans (it should accommodate the peaks in demands from the crystallizers with the continuous supply of standard liquor from the melter). These vacuum pans (Fig. 4) operate in batches, following a recipe with a set of stages using steam from the evaporation section to further concentrate the syrup until over-saturation stage. After discharge, the mixture of crystals and non-crystallized syrup (mother liquor) is stored in an agitated tank called strike receiver that feeds to another tank that supplies the mother liquor to a set of centrifuges where white sugar crystals are separated from the syrup. The centrifuges, that are modeled as a continuous component, use a small amount of water in their operation and produce two types of syrups (honeys): a high purity one, that is recycled to the melter, and a low purity one that is processed further in other stages (B and C) and finally is partly recycled to the melter too as a flow of low quality sugars (B and C). The recipes of the crystallizers are automated with Programmable Logic Controllers (PLCs) and the inner control loops are implemented with PIDs (level and pressure). In manual mode, the operator only decides when then recipe is started and the type of heating steam. Thus, the consumed steam and syrup and produced mother liquor by each crystallizer are not homogeneous and they depend on the stage (Fig. 5).

 ISBN: 978-1-61208-308-7

At the Sugar End, another three PIDS control the flow of massecuite (mother liquor), to the centrifuges and the levels of the melter and the tank that feeds the centrifuges (Fig. 3). However, the storage tank and the strike receiver levels are not controlled. Then, the high level operation of the process must guarantee that are kept within certain limits.

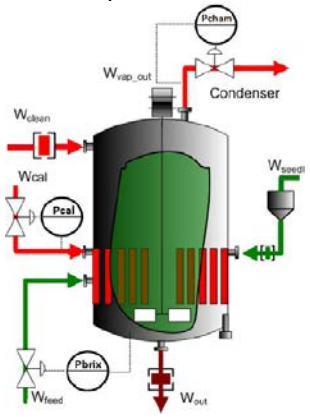


Figure 4. Sugar Batch Crystallizer

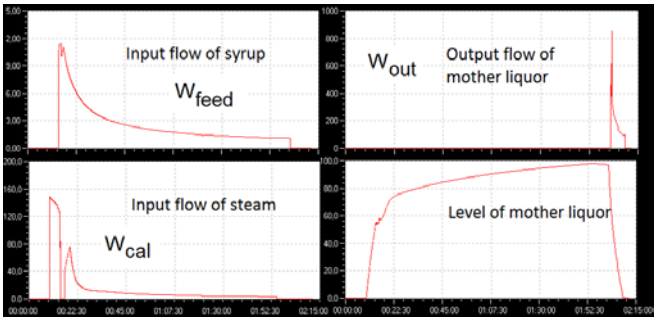


Figure 5. Flows (kg/s) and level (%) in a crystallizer cycle (time in hours: minutes: seconds)

D. Operation

The process operation is complex because both sections interact strongly in terms of mass flow and energy. For a given flow of juice before evaporation, the flow of produced syrup and the fresh steam demands depend on the set points for the fresh steam pressure and syrup concentration (Brix) controllers as well as on the steam demands from the crystallizers. On their side, each crystallizer's cycle time and its variable profiles depend on the concentration and purity of the evaporation syrup as well as on the pressure of the steam provided. Purity is defined as the % of sucrose of the solids dissolved in the syrup and affects also the % of crystals obtained in a batch. The cycle time determines the syrup processing capability of the Sugar End, which obviously limits the maximum allowable syrup flow from the evaporation. Other technical variable that affects the working of the Sugar End is the ratio water/massecuite in the centrifuge. Increasing it, the purity and flow of the rich syrup recycled are increased and it means that the cycle time of the crystallizers is decreased but, on the other hand, it decreases directly the white sugar crystals flow from the centrifuge. Thus, there are many choices of a set of key variables that determine the right operation of both sections, both from the point of view of the process working and its economy.

On the other hand, the crystallizers must be well sequenced. For instance, they mustn't start at the same time because it would imply a great initial demand of syrup and steam from the evaporation that it may not be supplied. Besides, the storage tank in the crystallization could empty and the strike receiver in the discharge stage could overflow. Fig. 5 shows the peaks in the steam and juice demand at the beginning of each cycle crystallizer and the peak in the discharge of mother liquor at the end of the same cycle).

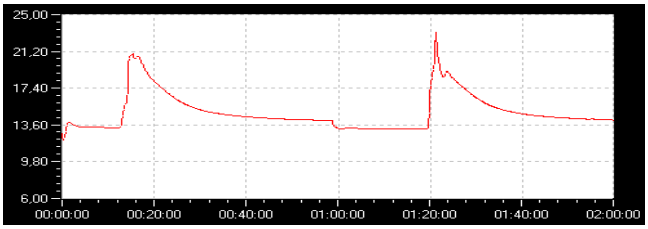


Figure 6. Fresh steam flow (kg/s)

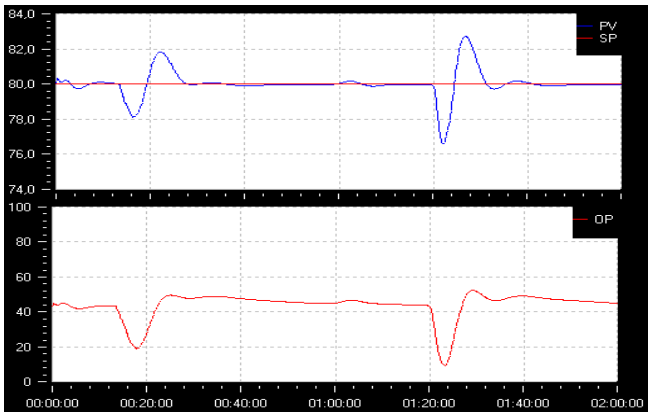


Figure 7. III effect level controller (PV: Process Variable (%), SP: Set Point, OP: Output to Process (%))

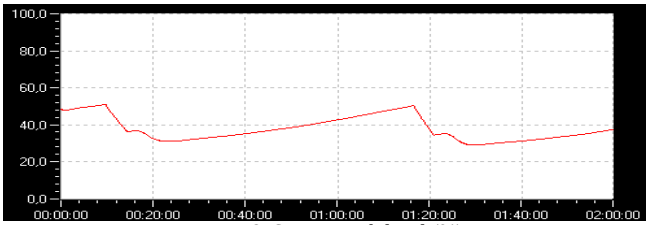


Figure 8. Storage tank level (%)

For instance, Fig. 6 shows the fresh steam demands when the crystallizers are well synchronized. The peaks, that are due to start a crystallizer, are homogeneously distributed along the time and their values are assumable by the boilers. Fig. 7 displays the level in the last evaporation effect and the control signal of the valve that govern the flow of syrup. This flow goes down when a crystallizer starts, because the increase of the steam demand of the crystallizer (see Fig. 5) affects to the Brix and pressure at the last effect that are controlled variables. Fig. 8 shows the level of the storage tank. When a crystallizer starts, it demands a great amount of syrup (see Fig. 5) and the produced syrup in the last effect decreases. Then, the level of this tank goes down. Later, this level recovers its value when the demand of syrup decreases and the produced syrup gets back its average value.

 ISBN: 978-1-61208-308-7

Summarizing, although the process is automated using PID and PLC controllers, some variables must be handled by a high level controller (or qualified operators) to assure that:

1. All the juice before evaporation will be processed, ensuring that the levels of uncontrolled buffer tanks are kept within certain limits.
2. Additionally, the process operation must minimize the energy consumption, trying to obtain a smooth and homogeneous fresh steam demand to avoid problems in boilers and turbo generators.
3. And, if possible, maximize the produced sugar.

To obtain these aims, the main decision variables are:

1. A set of set point controllers: (a) the fresh steam pressure to the evaporation from the boilers, P_{fs}; (b) the syrup concentration from the last evaporator, B_s and (c) the massecuite flow to the centrifuges, W_m.
2. The ratio water/massecuite in the centrifuge ($R_{w/m}$).
3. And, for each crystallizers, the scheduling, that is, when the operation of each batch cycle starts; and the selection of the steam source (steam I or II).

Finally, the control objectives must be met in the presence of disturbances. The main ones are changes in the amount of juice before evaporation (W_j), or its composition (Brix, B_j, and purity, Pu_j).

III. STUDY CASES

In this section, the study cases to test the different control algorithms are described. Each study case is characterized by some unknown operating boundary conditions and a process performance criterion.

A. Process Performance Criteria

The following criteria or targets have been defined:

First: Operate the plant to assure that uncontrolled buffer tanks levels (storage tank and strike receiver) are kept within certain limits and to maintain the value of the Brix and purity of the standard liquor within a maximum and minimum value due to technological requirements of the crystallizers.

$$L_{st}^{Min} < L_{st} < L_{st}^{Max}; \qquad L_{sr}^{Min} < L_{sr} < L_{sr}^{Max}$$
$$Pu_{sl}^{Min} < Pu_{sl} < Pu_{sl}^{Max}; \qquad B_{sl}^{Min} < B_{sl} < B_{sl}^{Max} \tag{1}$$

L_{st} and L_{sr} are the storage tank and strike receiver levels. Pu_{sl} and B_{sl} are the purity and Brix of the standard liquor.

Second: Minimize the energy to the system per kg of produced sugar (J_1, kJ/kg) and the variance of the normalized power to the system (J_2), respecting the first target.

$$J_1 = \frac{\int_0^T E(t)\,dt}{\int_0^T W_{sugar}(t)\,dt} \tag{2}$$

$$J_2 = \sigma^2 = \frac{1}{T}\int_0^T \left(\frac{E(t)}{\bar{E}(t)} - 1\right)^2 dt \tag{3}$$

T is the total simulation time. W_{sugar} is the produced sugar flow. E(t) is the instantaneous power or energy flow to the first evaporator, and, $\bar{E}(t)$ is the moving average of E(t).

$$\bar{E}(t) = \frac{1}{\Delta t}\int_{t-\Delta t}^{t} E(\tau)\,d\tau \tag{4}$$

Being Δt, the time interval for the moving average and it's equal to the total cooked time in one batch crystallizer (normally 9.000 seconds).

Third: Maximize the average profit per kg of produced sugar (J_3 (€/Kg)), respecting the **first** criterion:

$$J_3 = \frac{\int_0^T \left(\delta \cdot W_{sugar}(t) - \beta \cdot E(t) - \gamma \cdot W_j(t)\right) dt}{\int_0^T W_{sugar}(t)\,dt} \tag{5}$$

Where δ, β, γ are the prices of the produced white sugar in centrifuges (€/kg), consumed energy (€/kW) and juice before evaporation (€/kg), respectively.

Fourth: Maximize the average value of the flow of juice before evaporation (J_4 (Kg/s)), respecting the **first** criterion.

$$J_4 = \frac{1}{T}\int_0^T W_j(t)\,dt \tag{6}$$

This target is oriented to maximize the production capacity of the process, respecting the constraints.

In the simulation, these four indexes are calculated and their values are available for the users (Fig. 9).

PERFORMANCE INDEXES	
Average Steam Power per sucrose kilogram (J_1)	3987.42 kJ/kg
Average Steam Power variance (J_2)	0.000138
Average benefit per sucrose kilogram (J_3)	0.04 €/kg
Average Juice Flow (J_4)	40.29 kg/s

Figure 9. Performance indexes at the simulator interface

B. Operation Conditions

Two days was selected as the total simulation time of each exercise. The cooked time in a crystallizer is about 2 hours and a half, thus, the total simulation time must be broad enough to consider several batches. Thus, when the boundary conditions are modified, changing of a stationary point to another one spends several hours.

Two different boundary conditions are specified.

Operation condition one (OP1):

a. The system remains in a stationary state 6 hours.
b. Then, during 36 hours, the process is disturbed with variations in the flow, Brix and purity of the juice before evaporation.
c. Here, the initial values of the boundary conditions are restored and the simulation continues during 6 hours more until the end of the exercise.

Operation condition two (OP2): it is similar to the OP1, but the flow of the juice before the evaporation is not disturbed. Now, it must be managed by the high level controller.

Fig. 10 shows the trend of the uncontrolled level of storage tank when the operation condition one is selected and no actions are made over the process (red line). It can be seen that the storage tank level will go through the limits (20-80%). However, the black line shows the same situation with some changes in the crystallizer scheduling to satisfy the first target. Now, the performance of the uncontrolled tank level is better than in the previous situation and the constraint is satisfied.

 ISBN: 978-1-61208-308-7

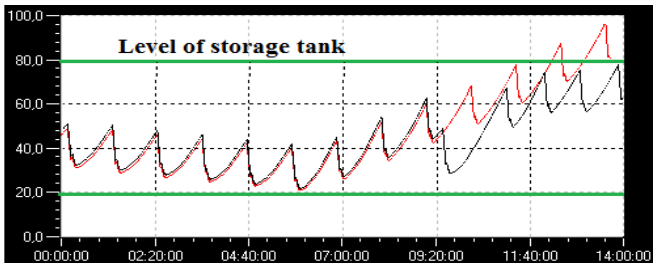


Figure 10. Red/black (without/with control actions

C. *Summary of Study Cases*

Thus, based on the operation conditions and the performance criteria, four benchmark study cases have been proposed (Table 1). Each study case is defined by one operation condition and one operation criterion or target.

TABLE I. *STUDY CASES*

	Operation		Variables	
Study Case	**Conditions**	**Criteria**	**Manipulated**	**Disturbance**
1	**OP1**	First	P_{sf}, B_s; $r_{w/m;}$ $W_{m;}$ Pans scheduling	W_j, B_j; $Pu_{j;}$
2	**OP1**	Second	Idem study case 1	Idem study case 1
3	**OP1**	Third	Idem study case 1	Idem study case 1
4	**OP2**	Fourth	Idem study case 1 plus W_j	B_j; Pu_j

The controller for the study cases number 1, 2 and 3, that isn't implemented in the software, should have the same data interface (Fig. 11). The disturbances are the same ones (flow, Brix and purity of the fresh juice) and the difference is the target function. The study case number 1 looks for a controller that operates the process subject to constraints in some variables without target function. The controller for the study case number 2 must operates the process subject to the same constraints and minimize the J_1 and J_2 indexes, that are related with the energy consumption. Finally, the controller for the study case number 3 must operates the process subject to such constraints and maximizes the J_3 index that is related with the greatest profit.

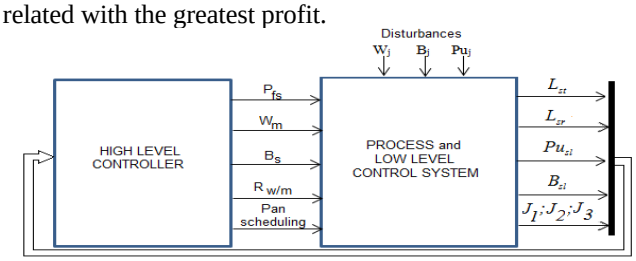


Figure 11. Control Structure for Operation Condition 1

For the study case number 4, the data structure of the high level controller changes, because the flow of the juice before evaporation is considered a manipulated variable, instead a disturbance. Now, the operation target to maximize is the number four (J_4). In this case, the controller must look for the maximum production capacity of the system with disturbance on the Brix and purity of the juice.

IV. SOFTWARE

The model and control system of the show case process is programmed in the simulation environment EcosimPro, which incorporates state-of-the-art simulation features. The model can be simulated, and the controller could be implemented, within the EcosimPro software environment but, for those that prefer to use other tools, and in order to facilitate the operation of the process from a graphic Human Machine Interface (HMI), a system with the architecture represented in Fig. 12 has been set up. It combines a real time execution of the simulation with a SCADA system to supervise and control the simulation. The communication between both elements is made by OPC. Additionally, using OPC, it's possible to connect the simulator and the SCADA with other external devices and software tools.

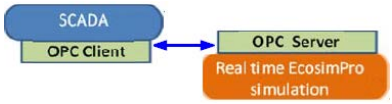


Figure 12. Show case software architecture

The SCADA system, called EDUSCA [19], was developed at the University of Valladolid and it's a not licensed tool that runs on PC over Windows OS. It has a friendly configuration environment and can work versus simulations or real process. It's used in a training simulator for control operators of sugar factories carried out by the Center of Sugar Technology and in some university labs. An example of the SCADA HMI is shown in Fig. 13.

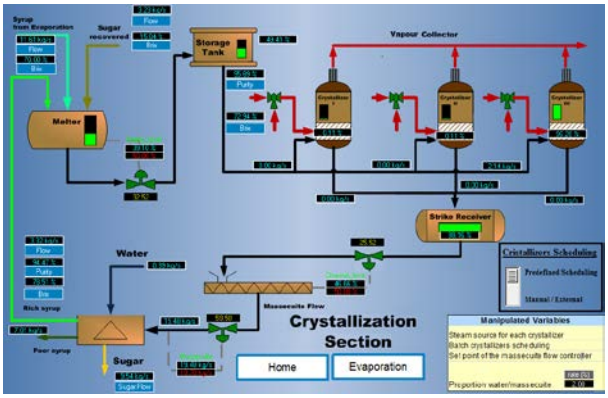


Figure 13. SCADA HMI, a synoptic of the Sugar End section

There are different display boxes associated to the process variables showing their values and units. Pushing on them, a graphic with the time evolution of the variable is displayed (some instances can be seen in Fig. 6-8). If the variable is a manipulated one of a control loop, full graphical information of the loop parameters and variables is provided. In order to facilitate the operation of the process, an alarm system is included, so if any variable goes beyond the allowed limits the process, the operator is warned with an alarm. An alarm register and historic data storage is maintained as well. The HMI includes the possibility of entering changes in previously defined modifiable process variables, set point controllers and scheduling of the crystallizers. By default, the simulator runs in real-time, the user can to speed-up its execution by a certain rate.

 ISBN: 978-1-61208-308-7

Besides, it is possible to interact with the simulator in different ways to the ones previously exposed (here, communication details are not given). In particular, it is possible to use the simulator from the MATLAB-SIMULINK environment. First, it is possible to connect the OPC server simulator to MATLAB-SIMULNK using the OPC toolbox, being MATLAB-SIMULNK a standard OPC client. Second, the EcosimPro simulation code, not the OPC server simulator, can be called directly from MATLAB using some functions provided for this purpose.

Summarizing, the user can access to the show case simulator using the next methods:

a) From the EcosimPro environment, if a program license is available.
b) EcosimPro simulation code can be used from MATLAB using a toolbox supported by EcosimPro.
c) It can be used the set formed by the OPC show case simulator and the SCADA system, that it is supplied by the developers.
d) The OPC show case simulator can be connected with any OPC client. It is possible to link it with several OPC clients simultaneously.
e) Especially, the OPC show case simulator can be connected with the OPC clients provided by MATLAB or SIMULINK.

The first two methods implies run the simulation as fast as possible, but in the last three methods it runs in real-time, being possible to speed-up its execution to a certain rate.

V. CONCLUSIONS AND FUTURE WORK

The paper has showed the result of one and a half year of work to develop a complex dynamic simulator of a standard industrial process. The simulator, that includes the low level control system, is thought to serve as a test bench of plant-wide control algorithms. The proposed control problem implies plant scheduling, operation and economic targets. Several scenarios and different performance indexes are defined. The combinations of these simulated scenarios with the targets constitute the so called study cases.

The simulation program can be used for different methods and tools. The MATLAB users can run the simulator easily. The simulator can run in real time or to a certain rate of real time, but in a standard PC, due to the model complexity, the acceleration rate is not greater than 20. In this way, a study case of 48 hours of real time can need about 2 hours of simulation time (the computational load of the high level plant controller is not considered).

The system can be used for the scientific community interested in the plant-wide control linked to economic requirements. Supplementary material and software of the sugar factory show case can be downloaded [20]. Researchers are invited to use this benchmark to test their control algorithms for hybrid complex systems.

As, strictly, benchmarking means comparing a solution to a reference, the exposed simulator cannot be still consider a benchmark, because a reference control system solution is not provided. Thus, the authors are working to supply a reference solution as soon as possible.

ACKNOWLEDGMENT

The authors are grateful to the European Union Seventh Framework Programmed [FP7/2007-2013] for its support under grant agreement n°257462 HYCON2 Network of excellence and to the Spanish sugar producers (ACOR and Azucarera Iberia) for their support to carry out this research.

REFERENCES

[1] HYCON2-Highly-complex and networked control systems [Online]. Available: http://www.hycon2.eu, [retrieved: july, 2013].

[2] R.Chylla and D. Haase. “Temperature control of semi-batch polymerization reactors”. Comput. Chem. Eng., vol. 17, 1993, pp. 257–264.

[3] G. Pellegrinetti and J. Bentsman. “Nonlinear Control Oriented Boiler Modeling–A Benchmark Problem for Controller Design”. IEEE Transactions on Control Systems Technology, Vol. 4, n 1, January 1996, pp. 57–64.

[4] U. Jeppsson et al. “Towards a benchmark simulation model for plant-wide control strategy performance evaluation of WWTPs”. Water Science & Technology, Vol 53, No 1, 2006, pp. 287–295, Q IWA Publishing. doi: 10.2166/wst.2006.031.

[5] ESA International. “EcosimPro User Manual, EL Modelling Guide”. EA International and ESA, 2011.

[6] F. Vazquez, J. Jiménez, J. Garrido, and A. Belmonte. “Introduction to Modelling and Simulation with ECOSIMPRO”. Ed. Pearson Educación, 2010.

[7] F. Cellier. “Continuous systems modeling”. Ed. Springer Verlag, New York, USA, 1991.

[8] P. Fritzson. “Introduction to Modeling and Simulation of Technical and Physical Systems with Modelica”. Ed. Wiley-IEEE Press, 2011.

[9] Modelica Association [Online]. Available: https://www.modelica.org, [retrieved: july, 2013].

[10] R. Mazaeda, C. de Prada, A. Merino, and L.F. Acebes. “Librería de Modelos Orientada a Objetos para la Simulación del Cuarto de Azúcar: Cristalizador Continuo por Evaporación al Vacío”. RIAI. Vol. 8, num. 1, 2011, pp: 100-111.

[11] R. Mazaeda, C. de Prada, A. Merino, and L.F. Acebes. “Hybrid Modelling of Batch Centrifuges as Part of a Generic Object Oriented Beet Sugar Mill Library”. Simulation Modelling Practice and Theory. Volume 22, March 2012, pp. 123-145.

[12] A. Merino, L.F. Acebes, R. Mazaeda, and C. de Prada,. “Modelado y Simulación del proceso de producción del azúcar”. RIAI. Vol. 6, num. 3, 2009, pp: 21-31.

[13] L.F. Acebes, A. Merino, R. Mazaeda, R. Alves, and C. de Prada. “Advanced dynamic simulators to train control room operators of sugar factories”. International Sugar Journal. Vol. 113, NO. 0000, October 2011, pp. 18-25.

[14] R. Mazaeda, A. Merino, C. de Prada, and L.F. Acebes. “Sugar End Training Simulator”. International Sugar Journal. Vol. 114, NO. 0000, April 2011, pp. 42-49.

[15] A. Merino, R. Mazaeda, L.F. Acebes, R. Alves, and C. de Prada. “Beet End Training Simulator”. International Sugar Journal. Vol. 114, NO. 0000, May 2012, pp. 34-40.

[16] OPC Foundation [Online]. Available: http://www.opcfoundation.org, [retrieved: july, 2013].

[17] R. A. Mc Ginnis “Beet sugar technology”. 3d Edition. Beet Sugar Development Foundation. Colorado, USA, 1982.

[18] P. Poel, H. Schiweck, and T. Schwartz. “Sugar Technology. Beet and Cane Sugar Manufacture”. Dr. Albert Bartens, 1998.

[19] R. Alves, J.E. Normey-Rico, A. Merino, L.F. Acebes, and C. de Prada (2005) “OPC based distributed real time simulation of complex continuous processes”. Simulation Modelling Practice and Theory. Volume 13, Issue 7, October 2005, pp. 525-549.

[20] HYCON2 WP5 sugar show case [Online]. Available: http://hycon.isa.cie.uva.es/home.html, [retrieved: july, 2013].

 ISBN: 978-1-61208-308-7

Towards Unified Conceptual Modeling and Integrated Analysis in Joint Applications of Project Management, Business Process Management and Simulation

Germano de Souza Kienbaum, Álvaro Augusto Neto
Associated Lab. for Applied Math. and Computing (LAC)
National Space Research Institute (INPE)
São José dos Campos, SP, Brasil
kienbaum@uol.com.br, prof.alvaro@uol.com.br

Carlos Alberto M. B. dos Santos, Andréa N. P. Durán, Renato Fernandez, Celso Israel Fornari
Department of Space Systems Eng. and Technology (ETE)
National Space Research Institute (INPE)
São José dos Campos, SP, Brasil
carlos.ambsantos@gmail.com, andrea@ccs.inpe.br, renato_fernandez@hotmail.com, celso@las.inpe.br

***Abstract*—This work proposes a systematic approach for model building and analysis of the product lifecycle processes of complex systems development, products and/or services, making use of Project Management, Business Process Management and Simulation techniques in an integrated and unified way. The approach is demonstrated making use of an academic model, describing an online Bookshop, but it envisages real systems applications and its use in Product Lifecycle Management in general. The modeling process starts with the creation of a unified reference process model, which is used for the development of multifaceted and cross consistent representations, each one related with a different view and discipline, aiming at the achievement of the complementary benefits resulting from their joint application.**

Keywords-unified conceptual modeling; product lifecycle management; business process management; project management; process simulation; process science and technology

I. INTRODUCTION

Process Science and Technology (PROST) is a designation given in [1, 2] to an emerging transdisciplinary science that addresses the integration and unification of concepts and techniques, which were originated and are traditionally used in several autonomous scientific areas, such as: Systems (Concurrent) Engineering (SE), Project Management (PM), Business Process Management (BPM) and Simulation.

The research scope of this emerging unified study area is the complete lifecycle of complex products and services: modeling, building, simulating, automating, managing and continuously improving the system´s concurrent engineering process, described as the integration of product development and organization management processes, by means of creating a unified methodology and its supporting tools.

The main focus of the unified PROST methodology is the development of a unique reference process model of the system´s engineering lifecycle (product development and organization management processes) under consideration and on its use to implement different views and applications, according to the various disciplines mentioned, in order to perform more complete studies and to achieve the complementary benefits of their joint application.

In comparison to Model Driven Engineering (MDE) [3] and Model Based Systems Engineering (MBSE) [4], one can say PROST advocates a similar approach, not by creating a "yet another modelling view", but by orchestrating existing methodologies, such as those used in SE, PM, BPM and Simulation, and unleashing their full potential in joint applications. The approach is the result of an ongoing research [1], but it has also been used in some real applications at INPE [2].

The current work describes the basics of the PROST modeling methodology, especially those aspects related with the Product Lifecycle Management (PLM), conducted with the simultaneous use of PM, BPM and Simulation modeling and analysis, and illustrates its application on a study case of an Online Bookshop. The SE dimension was intentionally not addressed in the work, partially due to the nature of the problem considered for demonstration, which does not require the use of proper product engineering procedures, but mainly for attending the article´s length restrictions.

This work is structured according to the following sections: Section II presents the unified approach for conducting joint modeling and analysis in PROST studies; Section III describes the problem used as study case; Section IV shows the creation of the reference model of the Bookshop Online system; Section V describes the PM model and its implementation; Section VI describes the BPM model and its implementation; Section VII describes the Simulation Model and its implementation; Section VIII discusses the integrated analysis and the results obtained with the application of the methodology; and Section IX draws some conclusions.

II. THE UNIFIED APPROACH FOR JOINT MODELING AND ANALYSIS IN PROST STUDIES

Fig. 1 shows the unified modeling approach for joint applications of the disciplines or dimensions of PM, BPM and Simulation proposed by the authors.

The rounded rectangles are the transformation processes and the cylinders stand for the model knowledge content (model representations) at a specific point in time.

 ISBN: 978-1-61208-308-7

The cycle starts with the definition of the system and of the study's objectives, which determine the scope of the model to be built. The Unified Conceptual Model is the main product of this phase. It contains the specification of the logical structure of the product lifecycle model together with the organization's management process (the unified system´s model according to the study's objectives). This phase also defines the system's boundaries, the model control parameters and eventual additional premises and restraints.

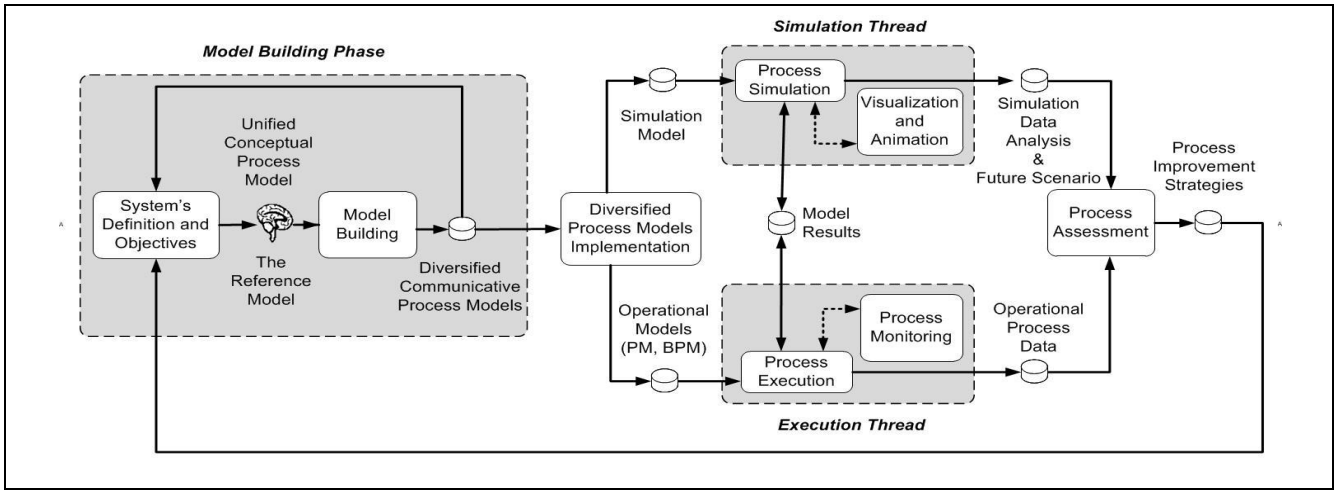


Figure 1. - The Unified Approach for PM, BPM and Simulation [11].

The Unified Conceptual Model is a concept that needs to be understood as the logical content of the system's operation taking into account the study's objectives. The concept of Unified Conceptual Model is similar to the one presented in Nance's conical methodology for building the conceptual model [5], which is defined separately and prior to the creation of the communicative process model. An equivalent of the Unified Conceptual Model in communicative format is designated Reference Model.

The Reference Model is created making use of Unified Lifecycle Modeling Diagrams (ULMD), a notation based on an extension of Activity Cycle Diagrams (ACD) and Project Evaluation and Review Technique (PERT) diagrams, originally proposed in Travassos with the denomination of Unified Simulation Modeling Diagrams (USMD) [6, 7]. The use of ULMD is a corner stone of the methodology, in order to assure the cross consistence of the other communicative models to be created later using several kinds of notations.

Besides the idea of including both the product´s development and the organization´s management processes (system´s processes), another important difference in relation to Onggo´s [8] formulation of multifaceted modeling is that all the representations are used to describe the same system, the integrated lifecycle processes of product and organisation, and they must be kept consistent with a single reference model, which is created at the very start of the modeling process.

The next step is building the system´s diversified communicative processes models, by transforming the reference model described in ULMD into different formats, such as PERT diagrams, Business Process Diagrams (BPD) and the simulation model.

The communicative process models undergo a third step of transformation, the implementation or model programming, yielding the implemented model or model's applications, which might be seen as different software systems or the same system that can be executed according to different threads. These threads are at least two: one for process enactment in production mode, with business process and project management functionalities; and the other one for simulation with design of experiments, the building of scenarios, assessment analysis and results displaying functionalities embedded. Both threads are fed by the process models, produced from the set of communicative models, all verified to assess their consistency and validity in regard to the unified system's specifications. Data collected during real system's operation are used as input data for simulation model execution, making validation easier and future scenarios projections more reliable.

The results from the different threads of execution (project management execution, business process management execution and simulation) provide information for the next phase of process analysis and assessment. The process analysis and assessment step shall be carried out according to the diverse views and disciplines, making use of the appropriate metrics, with the aim of continuous process model improvement, by restarting the cycle.

III. THE ONLINE BOOKSHOP PROBLEM

The study case selected to demonstrate the application of the methodology proposed in this work is one of a hypothetical online Bookshop, as presented by Aalst [9].

The process model of the virtual Bookshop might be decomposed in three different sub-processes, each one corresponding to a different class of entity or participant of

 ISBN: 978-1-61208-308-7

the process: the Clients or Customers, the online Bookshop itself and the Publisher(s).

The clients access the Bookshop´s site online via the Web. Initially the client places an order for a book filling out a form and his order and personal data are registered by the system. The Bookshop then sends the client´s request to a Publisher, who will check if the book is available in stock. The publisher sends a message with the information requested and, if the book is not available, the bookshop online communicates this fact to the client and the process ends. If the book is available, the online Bookshop informs the client and pays in advance the publisher, who sends the book directly to the client and notifies the Bookshop of the fact. The Bookshop sends the invoice to the client, who then pays the Bookshop, and the complete process is finished.

IV. UNIFIED CONCEPTUAL MODELING AND THE REFERENCE MODEL

The reference model is created at the start of the modeling process, making use of the ULMD notation, and it is used to maintain the consistency of the PM, BPM and the Simulation models, to be created later, making use of the appropriate tools. Fig. 5 placed at the end of this work shows the reference model of the online Bookshop problem.

The main entities involved are: the Customers, the Bookshop and the Publisher, with the darker grey color in the middle used to differentiate the Bookshop from the other entities, placed in the above order. The squares are the macro processes or single activities (transformations that require real time to be executed) and the circles stand for the queues or actual location in which each of the entities (or token representing the control flow) stay along their pathway in their process lifecycles. Actually, one could think of these locations as databases or knowledge contents carried by the entity at a specific point of its path and the complete set of these databases as the descriptive (structural) model, whereas the process map shows the dynamical model associated with the product evolution along its lifecycle. If a complex product is under construction, one could think that a complete representation (including functional aspects of the product, using for example SYSML [10] notation) could be an artifact produced by a transformation activity in some point in time along the workflow of the product development process.

This type of diagram shows important aspects of the model logic, such as: the main entities which are involved, the flow of control and how the individual processes communicate with each other, the queues in the system, and which resource is responsible for the execution of each activity. In the case of Customers, just one individual at a time is responsible for the activity being executed in his process lifecycle, but the other entity lifecycles could have many resources associated with them, which would mean that several instances of an order or other kind of entity flowing through the process map could be processed simultaneously. The resources are part of the organization´s asset and they might have an associated utilization cost, as well as their availability could be established according to a schedule varying with daytime or weekday, for example. The quantity of resources of each kind can be fixed based on some kind of cost consideration or the throughput of the system can be chosen as the primary control variable for process optimization purposes, and the workload and number of resources necessary could be derived – making use of simulation modeling and performance analysis.

V. THE PROJECT MANAGEMENT MODEL AND ITS IMPLEMENTATION

The traditional way to describe a project is by representing it as a sequenced network of activities, by means of diagrams known as PERT [11], a renowned and well documented technique, used for management of engineering projects, be it a service provided by an enterprise or an industrial product, aiming at their planning and execution control. Fig. 2 shows the PERT network of activities for the Bookshop Online Problem.

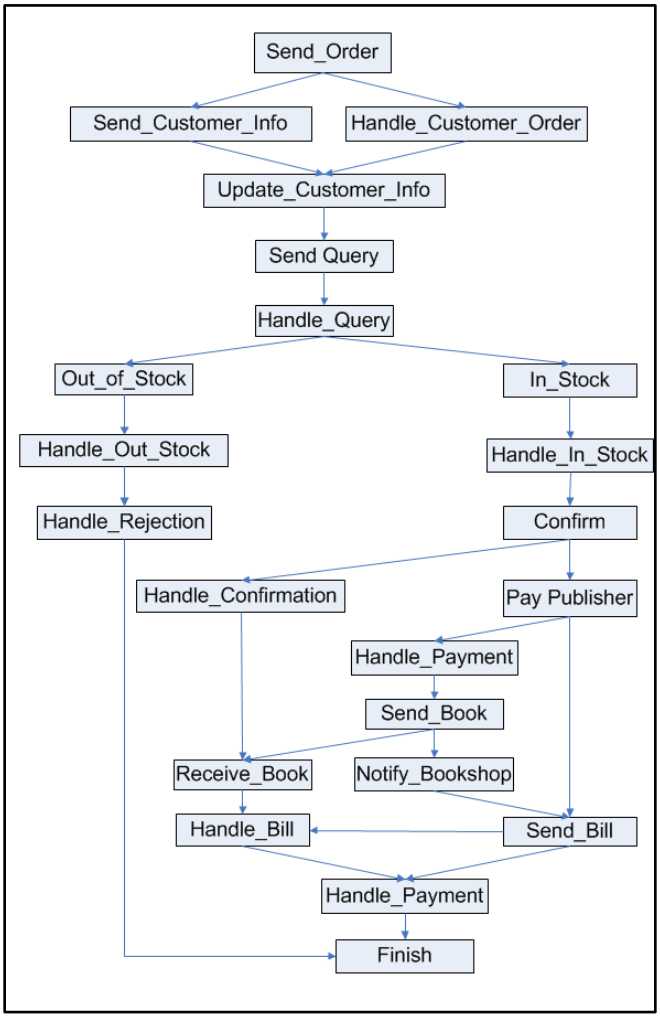


Figure 2. Project Management Model of the Bookshop Online.

A project is traditionally seen as “a single enterprise, of limited time duration, formally organized, which aggregates and applies resources aiming at the fulfilment of precisely pre-established objectives” [12]. This "single enterprise" way

of viewing projects might be the reason why projects have been traditionally treated in the literature of the area and by computer systems developers as a matter completely dissociated from business process management and industrial serial processes.

The analogies between these study areas and their particular types of problems become evident, however, when one considers a project not as a single process, but as a serial one or, equivalently, when one looks at it as a multi-project, made by the repetition, in parallel and possibly with some delay between each start, of its basic single process. The objective of a project management study could then be seen as the determination of the ideal basic process descriptive of the project, corresponding to the optimized distribution of all allocated resources to achieve the best performance both in terms of total process time and cost through all stages or phases of the project execution.

These models can then be used for the analysis and management of the project under development, making use of PM supporting systems, including the identification of the critical path, the scheduling of all activities, the allocation of all resources, the assessment of time duration and costs of partial segments (model´s components) or total project´s process model.

VI. THE BPM MODEL AND ITS IMPLEMENTATION

A business process occurs when different entities (individuals and/or organizations) interact to achieve a common business goal. The business process model is described by a workflow of activities, that is, how the entities interact to perform certain tasks in order to meet the business goals. The jobs under execution flow among them and each entity performs the part of the business process he/she is responsible for.

The BPM study area requires process model building in Business Process Management Notation (BPMN) as a means for the creation of representative models of the product´s or services´ development processes provided by an organization, in order to better understand them and to allow their continuous improvement.

Fig. 3 shows the transformation of the ULMD reference model into the BPMN representation using the Bizagi´s Process Modeller graphical editor.

Bizagi is a Business Process Management System (BPMS), a system used for implementation of solutions in order to model, analyze, manage and improve performance of an organization´s business process [13].

Bizagi provides a graphical user interface for business process design based on the BPMN notation – called Process Modeller – and a Suite Bizagi, an environment with many functionalities, for the implementation and the deployment of applications to help real system´s operation, automation, management and control, the monitoring of results and the analysis of performance, in order to continuously improve the organization´s business process.

Graphical editors like Process Modeller allow model building of business process operations making use of a network of graphical objects, constituted mainly by the activities, the routing and the synchronization gates, and the lines showing the flow of control - the activities´ sequence of execution. A BPMS, like Bizagi, provides several functionalities to help the development of automatic BPM applications, such as:

- Model building, workflow application generation, execution, control, management, automation and simulation of business processes;
- Real time monitoring;
- Communication and quality improvement of business processes;
- Increasing of efficiency and productivity;
- Fast results and good ROI.
- Optimization and continuous improvement of business processes at low cost;

The editor presents a drag and drop menu with the BPMN elements for rapid prototyping of new models, as well as the maintenance or reuse of existing ones or parts thereof. Each component can be individually entered and configured, making use of the templates representative of the BPMN elements and of context sensitive help, allowing the user to speed up model construction.

The Customer, Bookshop and Publisher processes shown in the reference model were transformed into pools in the business process model and a one-to-one correspondence of each entity´s activities was kept in the BPMN representation format, as well as were kept the control and messages flows connecting the activities in each pool representing the entity´s process cycle.

An interesting remark shall be made in relation to the apparent duplication of the UPD_C_Info activity, which was originally depicted as a single activity under responsibility of the bookshop agent executed in cooperation with the Customer agent in the ULMD model, and now is depicted as a send-receive message pair in the BPMN representation. Actually, this kind of coupled or synchronized activities shown in the ULMD model is very common and they have been already explicitly represented in other parts of the model, such as the pairs (S_Query/H_Query), (In_Stock/ H_In_Stock), (Out_of_Stock /H_Out_of_Stock), (Confirm/ H_Confirmation), (Pay_Publisher/H_Payment) and (S_Bill/ H_Bill). These coupled activities can either be represented as joint tasks performed under the responsibility of a single agent or split in different pools, in which case they are represented as pair of send/receive sequential activities linked by messages. They are executed in a synchronized way, in the sense their agents interact during their execution and are liberated to continue their individual lifecycle processes after these activities are finished. Both kinds of representations reflect a similar logical construct and they were used interchangeably in the implementations of the BPD described in this section and in the simulation model to be described in Section VII of this article.

 ISBN: 978-1-61208-308-7

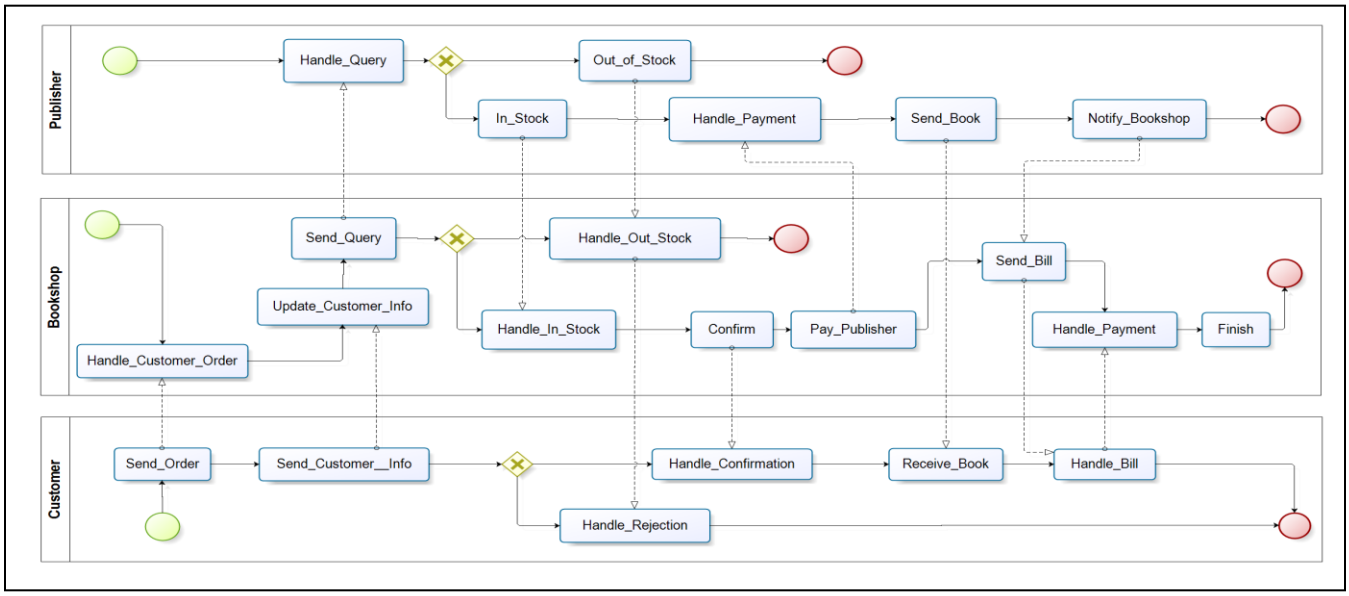


Figure 3. BizAgi´s Model of the Bookshop Online

The initial BPMN model description built with Process Modeller can be later developed into a fully operational application making use of the BizAgi Suite module. For example, certain activities might require a form for data input by the user. These forms can be designed by the application developer and local and global variables can be defined for model parameterization and flow of control description. Databases can also be used linked to the model operation. There are several objects which might be configured for expressing different pre-built tasks and a report mechanism linked to the databases helps the management and analysis of model operation.

The BizAgi Suite functionalities allow the model to be deployed and put into operation. The use of this additional module allows the deployment of the model in the form of a Web application accessible by usual Web browsers. A workflow execution engine provides the enactment of the activities to be performed by the agents in the application created to support the management of the real system.

VII. THE SIMULATION MODEL AND ITS IMPLEMENTATION

Similar to the BPMN model representation created with the BPMS BizAgi, a simulation model is built based on the reference model. In the absence of automatic mechanisms for model transformation and verification, this procedure requires the modeller to check himself model fidelity with the reference model and its overall consistence. The transformation from the ULMD notation into a workflow of activities to be implemented using the graphical elements of Simprocess is done in a very straightforward way. One can keep the one-to-one correspondence of activities, as mentioned in the case of the BPD shown in section VI, but the representation chosen was in the form of macro or coupled activities, as a way to emphasize their equivalence. Fig. 4 illustrates the online Bookshop model implemented with the Graphical User Interface (GUI) of the Simprocess simulation system [14].

Simprocess is a tool for hierarchical process simulation modeling that combines workflow modeling with discrete event simulation capabilities and Activity Based Cost (ABC) analysis in a single environment with a friendly GUI for process model design. The model built using Simprocess shown in Fig. 4 has six types of entities defined, namely: Customer, Bookshop, Publisher, Message, Book and Payment. The entities Customer, Bookshop and Publisher are the same defined in the ULMD model. In the original Simprocess GUI interface, the drawing has colors, which help differentiate the entities, but the connections were also named to facilitate their identification in tons of grey: Customer, Bookshop and Publisher are represented by continuous lines, depicting the flow of control or pathways of the entities, and the dotted lines represent the message exchange between the processes. The dashed lines indicate pathways of more than one class of entity, for the sake of simplifying the graphical model representation, although these entity flows could be duplicated and differentiated individually, if preferred. The kind of entity flowing in each connection is also identified by text boxes associated with this particular conection.

The entity instances of class Message are created or have their type transformed whenever a process needs to send a certain kind of message, for example at the moment the client enters the system and fills out a form (book order), when the bookshop sends this order further to the Publisher (query) and when the Publisher answers the request for information (book in or out of stock). The entities Book and Payment have a similar nature to the Message entity and they were created to represent the pathway of the Book and of the Payment followed in the system, respectively. Additionally to these entities, there were two types of resources created, called Bookshop Resource and Publisher Resource, to allow

 ISBN: 978-1-61208-308-7

multiple executions of the activities performed under the responsibility of the entities Bookshop and Publisher.

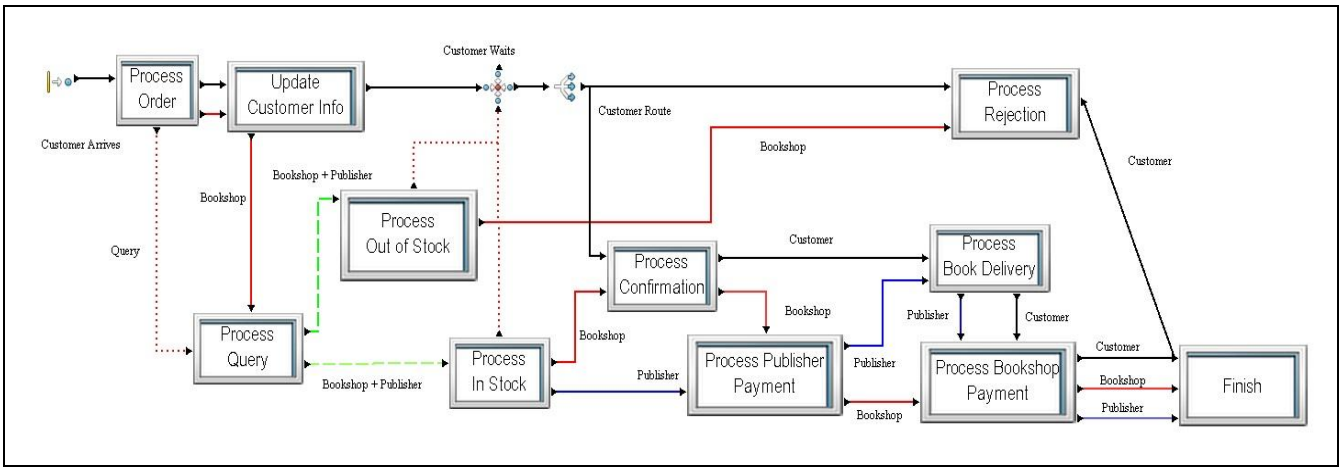


Figure 4. Simprocess Model of the Bookshop Online

The system was tested for gradual workload increase and determination of warm up period. The warm up period, the time necessary for the system to reach the steady state, was determined based on the quantity of clients being processed in the system. Some pre-built standard graphical displays can be activated in Simprocess to help performing these kinds of analysis.

The system maximum workload was determined by varying the quantity of available resources of each type in different simulation runs. Initially the resources were defined independently for each activity and just one instance of each type was made available. The quantity of available resources was gradually increased until there were no more entities waiting for resources in the queues in front of each activity and the maximum total number of occupied resources was reached. Based on these numbers new simulation runs were executed with just two kinds of resources defined, as previously described: Bookshop Resource and Publisher Resource.

The total number of resources needed in this last case is a bit lower than the total sum of the individual resources previously found. This is due to the fact that when two types of resources are used they can be assigned in different points of the process lifecycles of the entities being served, freeing some bottlenecks. The use of individual resources in each activity might result in an oversupply, because individual resources cannot be allocated elsewhere in the process lifecycle of the entities, helping the process overall throughput.

Many other predefined variables for model assessment maybe used, besides the total number of clients in the system, such as the quantity of clients being processed or waiting in specific points in their process flow (activities or queues, respectively), as well as the total number of busy resources and the percentage of the time they have been busy.

BizAgi and Simprocess have no special integration mechanisms, except for import/export using XPDL model formats, which are not fully compatible. The integration of these kinds of systems is a trend though and Simprocess´ manufacturers advertise on the existence of integration facilities with Ultimus BPMS and with MS Project, which makes it an interesting choice for exploring the unified conceptual modeling concepts proposed in this work.

The most updated version of BizAgi Process Modeller, Version 2.5 [13], presents also some functionalities for the execution of simulation runs, but these functionalities do not substitute the use of a simulation system like Simprocess [14], because the last one has pre-built facilities for the creation of more complex models, for making them more faithful to the real system and for allowing project of experiments and an adequate performance analysis, making use of scenarios.

VIII. Integrated Analysis and Assessment of Results

This Section presents an integrated analysis and assessment of results obtained from the application of the PROST methodology to the case study, divided into the following topics: Domain of applicability and limits of the approach; Benefits of the methodology and its tools; and actual state and future goals of the research.

A. Domain of Applicability and Limits of the Approach

The domain of applicability of the proposed approach for Unified Conceptual Modeling and its application in PLM of complex product and service development is given by the discrete event systems which can adequately be represented by the ULMD representation, which is a hybrid creation from ACDs and a workflow or PERT-like network of activities.

This problem class tends to be very large because a network of activities is a good representation of processes executed in discrete event systems in general. An additional remark to this point is that the use of this kind of diagram in PROST studies is solely intended for the product´s lifecycle process modeling, whereas other kinds of diagrams, such as SysML, are kept for other static and dynamical product

 ISBN: 978-1-61208-308-7

descriptions, making the overall modeling procedure very powerful.

ULMD is essentially a sub-set of BPMN; nevertheless, it is necessary as a unified conceptual/communicative modeling notation due to its higher level of abstraction, aiming at allowing flexible modeling with the minimum number of elements. The kind of problems that might be represented by the ULMD notation includes even discrete event systems with a cyclical nature, such as a serial production processes. The transformation from a cyclical nature into PERT-like networks is possible and demonstrated in [7], since the main path and its ramifications do not need to correspond to one of the real entities processed by the model. It might be described by a virtual entity "execution orders or flows of control" that splits for branches that are executed in parallel or return for recycling in case it proves necessary, in the same way one can use BPMN for describing complex processes.

The application of the methodology and tools derived from the project management area is based on the idea that the complete production process or its segment currently under analysis can be seen as a single project. The successive cycles representing the different batches of products are dealt with by replicating the basic process, which might be restarted any number of times, with or without a time delay, creating a network of activities whose graphical representation is drawn and executed sequentially from left to right or top to bottom.

Serialized production processes are therefore represented as equivalent to a complex process/project of a multi-project nature, made of several instances of the single process, each instance initialized with a different start time.

There is no need to consider multi-projects with a high number of identical processes, because the finish time of the first process would limit the number of total processes which would be simultaneously active in the system. The system's steady state behaviour would thus depend only on the number of simultaneous processes being carried out in it at any one moment.

In some cases, it might be necessary to repeat some parts of a process to create its complete graphical representation, if the same entity needs to repeat a sequence of activities for a fixed number of times, differently of the treatment described above for cycles that are originated from the processing of successive jobs or entities.

A problem arises when the number of times a segment must be repeated is dependent on a variable attribute for different instances of a class of entities being processed in the model. In this case the process cannot be described in this level of detail as a fixed PERT-like network of activities that needs to be traversed only once by that entity class or transaction existing in the model.

In these cases, the problem may only be described as a PERT-like network of activities if the level of detail is reduced, that is, if the problem is modelled in a higher hierarchical form, with some details being encapsulated into a macro activity that has to be considered as a single activity for the purpose of complementary time and cost analysis proposed by the approach.

B. Benefits of the Methodology and its Tools

The idea is to take advantage from the joint application of several modeling and analysis techniques in support of the Product Lifecycle Management of complex products´ and services´ development processes, in order to benefit from the complementary aspects for which each kind of these techniques is especially stronger.

From the project manager standpoint, it is expected that the application of BPM and simulation into PM will complement the benefits from the isolated application of the PM technique. BPM applications can be built to automate and help the management process. Project assessment will be made by a combination of the normal procedures used in project management with the addition of the simulation technique, with the aim of enhancing the understanding of the factors and strategies which significantly affect project execution.

The analysis, using simulation, of multi-projects made by several single projects of identical nature, will produce a better understanding of its characteristic single project or process and allow the improvement of its descriptive process, by optimizing resources allocation and shortening the complete process or segments execution times, while keeping control of activities costs.

The optimization will be based on the dissociation of the time delay incurred by the entities staying in the queues in front of each activity from the proper duration of these activities, what is treated as an aggregated estimation in the project management current studies, based on conservative estimative. The reduction of these waiting times by increasing the number of resources allocated, while keeping control of their relative costs, shall produce on its own a major gain of productivity in the execution of single projects.

The gain in productivity will be even greater when one considers the scaling factor, existing in systems in which real multi-projects or multi-processes need to be carried out, with their start time shifted only by a certain delay and their processes being executed in parallel, by big work teams divided in classes by their specialities.

The lack of this kind of analysis in project management studies actually performed is explained by the fact that the existing software tools used in this study area have no capabilities for experimentation of alternative forms for the modeling of their processes, for the animation of the passage of time, and for the testing of its dynamical resources allocation in the case of multi-projects. These are clear deficiencies of these systems, when they are compared to the existing simulation systems. These mechanisms will be an essential part of the hybrid PM, BPM and simulation environment here proposed.

Simulation studies performed with this hybrid environment will keep track of the complete map of dependences and sequencing of all activities, as well as of the resources allocated in the model. Experimentation and simulation model assessment will be improved and productivity will be enhanced in some segments or in the overall project´s lifecycle, through the optimization of

 ISBN: 978-1-61208-308-7

resources allocation and the minimization of completion times, subject to costs constraints.

This result can be achieved by creating pre-built mechanisms that are independent from the specific model under consideration, allowing model assessment for project enhanced productivity to become part of the normal objectives of simulation systems. These model independent mechanisms may be developed by using existing functionalities, or may be newly created if these functionalities are still not available, in an integrated PM, BPM and simulation environment.

C. Actual State and Future Goals of the Research

The ULMD model of the online Bookshop was implemented both in BizAgi and the Simprocess simulation systems. These implementations were conducted by groups of research students as final simulation course projects. The choice of the application systems above was made solely due to their availability as course material, but any existing PM, BPM and process simulation software available in the market can be used for this exploratory phase of the methodology development.

Concepts such as the identification of idle times of entities staying in queues in front of activities and dynamical resources allocation via the use of simulation were applied to reduce segments or overall process completion times and costs in the model. Concepts such as critical paths and completion time for a segment of the process, typical of project management technique, were not applied in the study, since the course´s objective was the joint application of BPM and Simulation techniques only. Efforts in this direction can be proposed for future research, with the goal of yielding greater productivity and a thorough analysis of possible strategies for system's operation.

As expected, the tools chosen for implementation showed their deficiencies in dealing with some aspects of the modeling, such as replicating the processes and allowing the conduction of experiments with multiple processes in the case of simulation and with making activity duration dependent upon the quantity of resources of each class allocated in the model, in the case of the BPM tool.

IX. CONCLUSION

The joint use of PM, BPM and simulation in process modeling and analysis reveals that they have a complementary nature. The first two of these procedures allow for a better understanding of the logic and strategies for managing the lifecycle of the entities flowing through the system. The last one allows for the analysis of the dynamics of their processes, including optimization of resources allocation, a better evaluation of completion time of partial or complete production cycles, as well as their cost assessment. The combination of these techniques is therefore very promising, but the advantages of their joint use have not been exploited, as far as the authors are aware, for two main reasons: (1) first and more importantly, because there is no unified conceptual modeling methodology, capable of unifying the modeling procedure prior to the application of these individual techniques; (2) second, because they have been designed with different purposes and knowledge basis in mind, without considering their complementary nature.

The first aspect can be dealt with by developing a unified/integrated conceptual modeling methodology, for which one hopes this work may have contributed, but the dream of achieving the full benefits of a unified methodology shall only come totally true if one undertakes the design and the building of a hybrid PM, BPM and simulation environment to deal simultaneously in a unified and integrated way with all issues involved in these autonomous and complementary study areas.

This work addresses the identification of the similarities and differences between model representation formats used in the different disciplines dealing with process modeling and the formulation of concepts and procedures for their integration. An initial conceptual PROST framework has been proposed, which one hopes that will lead to the development of a complete methodology and its supporting tools to deal with the issue of improving PLM procedures. The application of PM, BPM and simulation performed on the study case of a Bookshop Online illustrated further the use of the methodology under construction. The continuation of the development and application of the methodology will require the use of existing PM, BPM and simulation systems to perform several case studies, as well as the creation of a new hybrid simulation environment, which on its turn will require quite a lot of software development effort.

ACKNOWLEDGMENT

To the Brazilian National Research Council (CNPq) for sponsoring this research on Process Science and Technology with a postdoctoral research grant from the Program Science without Borders for the main author. Thanks to all research students from the CSE-326-4 Course on Simulation Modeling and Business Process Management, taught in the first period of the year 2013, as part of the ETE/INPE Post graduation Program in Systems Engineering and Project Management, for their contributions to this work´s models.

REFERENCES

[1] G. S. Kienbaum, L. A. Silva, F. Loureiro, A. Augusto Neto, and S. Robinson, "A Framework for Process Science and Technology Applied to Concurrent Engineering", Proc. of the 19th ISPE International Conference on Concurrent Engineering (CE 2012), Springer-Verlag, September 2012, vol. 2, pp. 1033-1044.

[2] L. A. Silva, G. S. Kienbaum, G. Loureiro, and M. M. Tanik, "A Process Science and Technology Study Applied to the Laboratory of Integration and Testing of the National Space Research Institute (LIT/INPE)". Proc. of the Society for Design and Process Science Conference (SDPS 2011), Jeju Island, June 2011.

[3] J. Bezivin, "Model Driven Engineering:An Emerging Technical Space".R. Lämmel, J. Saraiva, and J. Visser (Eds): GTTSE 2005, LNCC 4143, Springer, 2006, pp. 36-64.

[4] J. Estefan, "Survey of Candidate Model-Based Systems Engineering (MBSE) Methodologies", International Council on Systems Engineering (INCOSE), May 23, 2008. INCOSE-TD-2007-003-02. Available in: http://www.incose.org/productspubs/pdf/techdata/mttc/mbse_methodology_survey_2008-0610_revb-jae2.pdf, Retrieved: September, 2013.

 ISBN: 978-1-61208-308-7

[5] R. E. Nance, "The conical methodology and the evolution of simulation model development". Annals of Operations Research, no. 53, 1994, pp. 1-45.

[6] P. R. N. Travassos, "An Integrated Approach for Business Process Management and System Simulation and its Application in Project Management", PhD Thesis, National Space Research Institute, São José dos Campos, 2007. Available in portuguese in: http://urlib.net/sid.inpe.br/mtc-m17@80/2007/06.12.18.51 Retrieved: September, 2013.

[7] P. R. N. Travassos, G. S. KIENBAUM, "Methodologies and Tools for the Integration of Process Simulation and Project Management". Proc. of the VII Navy´s Symposium on Operational Research and Logistics, December, 2004, CASNAV,vol. 1, pp. 150-163.

[8] S. B. Onggo, "Towards a unified conceptual model representation: a case study in health care". Journal of Simulation, vol. 3, no. 1, 2009, pp. 40-49.

[9] W. van der Aalst, M. Weske, and G. Wirtz, "Advanced Topics in Workflow Management". Journal of Integrated Design and Process Science, vol 7, no. 1, March 2003, pp. 49-77.

[10] SysML, Systems Modeling Language User Guide, Available in: http://www.sysml.org/, Retrieved: September, 2013.

[11] Project Management Institute (PMI). "Practice Standard for Scheduling", 2nd ed. Project Management Institute, 2011, pp. 9-12, pp. 35-39.

[12] Project Management Institute (PMI), "A Guide to the Project Management Body of Knowledge (PMBOK® Guide)", 5th ed. Project Management Institute, 2013.

[13] Bizagi, Bizagi Process Modeler User Guide, Available in: http://help.bizagi.com/processmodeler/en/, Retrieved: September, 2013.

[14] CACI, Simprocess Product Overview, Available in: http://simprocess.com/products/products.html, Retrieved: September, 2013.

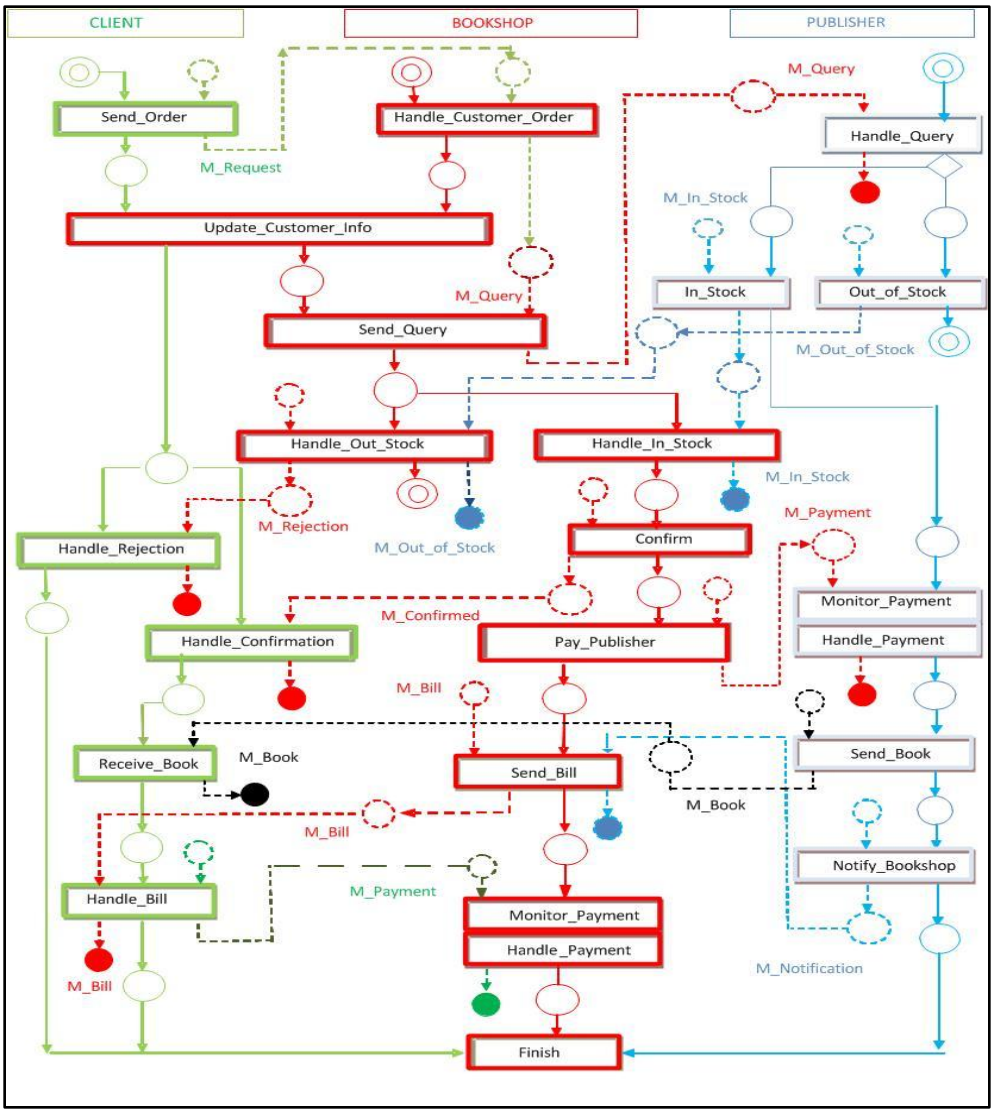


Figure 5. The Online Bookshop Reference Model.

Application of Lean Thinking Using Simulation Modeling in A Private Hospital

Ayman Tobail[(*)], Patricia Egan[(^)], Waleed Abo-Hamad[(*)], Amr Arisha[(*)]
[(*)]3S Group, College of Business, Dublin Institute of Technology, Dublin, Ireland.
[(^)] UPMC Beacon Hospital, Sandyford, Dublin 18, Dublin, Ireland.
Email: ayman.tobail@dit.ie, Patricia.Egan@BeaconHospital.ie, waleed.abohamad@dit.ie, amr.arisha@dit.ie

***Abstract*—Timely access, prompt responses to patient needs, and availability of resources to deliver quality service are the key priorities of healthcare systems, in particular hospitals. To cope with these constraints, healthcare managers have turned into lean thinking and approaches in their attempts to reduce non-value added activities and save costs by reducing wastes. This paper presents a case study of a private hospital in Dublin that used integrated approach of value stream mapping and simulation modeling to assess lean implementation in admission and discharge processes. Simulation enabled the strategic management to examine the outcomes of three possible improvement scenarios on hospital performance before implementing lean strategies. The proposed methodology helped to identify bottlenecks and non-value added procedures. Results analysis showed potential improvement in patients' admission and discharge cycle times, and offered the hospital the cost saving opportunity of reducing the numbers of bed required.**

Keywords-Lean; Modeling and Simulation; Healthcare.

I. INTRODUCTION

Healthcare resources, like those in any other industry, are in high demand, and the current economic climate challenges both public and private hospitals to contain costs whilst optimizing their use. [1] outlines the challenges that face healthcare: slow economic recovery, rising costs and reduced rates of reimbursement by insurance companies, to name a few. Whilst their focus is on the US healthcare economy (which is primarily private) the issues faced are applicable worldwide. In Ireland, the last twenty years or so (pre recession) has seen an increase in people's earning and subsequent purchasing power, with a consequent rise in the percentage of the population with private health insurance, to around 46 per cent of the population [2], which equates to 2.25 million private healthcare policy holders. However, this has led to a dichotomy: far from reducing the pressure on public services, the upsurge in private health insurance and the aging population profile have brought new demands for services in both public and private sectors. So, the number of private providers responding to this need has also increased over the last decade or so. Whilst healthcare is seen as relatively recession proof, the prolonged economic recession has impacted on all healthcare providers, increasing expectations on health service providers to raise service efficiency from external entities such as the government, the public/patients and, of course, insurers. Whilst such efficiencies have always been the focus of the private sector, in recent years the requirement has been expressed with renewed vigor. When dealing with challenges such as reducing costs, a hospital must take a strategic medium- to long-term view on how to proceed to how best to serve all its stakeholders fairly. Such strategic decisions determine how the organization will align itself with its environment [3].

To ensure the best possibility of survival, organizations must scan the horizon for opportunities and capitalize on those that exploit their core competences. In the private sector, companies it must scan the horizon for opportunities, and capitalize on those that best exploit their core competences to give them the greatest chance of survival. These opportunities may include using frameworks and tools from other industries - and such tools can be adapted and developed to help healthcare organizations address their challenges. Lean thinking and simulation modeling offer two distinct frameworks which organizations can use to streamline their processes.

In the rest of the paper, Section 2 reviews related work. Section 3 introduces project background. The proposed methodology is presented in Section 4, followed by experimentation and analysis in Section 5. Section 6 presents limitations and future work, while Section 7 concludes the paper.

II. LITERATURE REVIEW

Lean healthcare is the philosophy of improving flows of patients, information or goods by eliminating waste from the process [4] through 'understanding current processes, identifying the areas for improvement, and implementing necessary change' [5]. The Lean approach seeks improvements within the organization's existing processes but without the substantial reorganization that would require costly investments. Waste erodes quality and results in 'inefficiencies, higher operating costs, increased potential for errors and worker frustration' [6] – so, the logic behind lean thinking is to pursue the optimization of value streams (from the consumption point of view) by eliminating waste and non-value added activities. To identify the sources of such waste and non-value added activities, as well as opportunities of improvement, value added activities must be mapped using systematic tools and techniques [7]. A value stream can be defined as the collection of activities that are operated to deliver a product or service or a combination of both to a customer [8]. The Value Stream Map (VSM) technique demonstrates material and information flow, maps out value-added and non-value-added activities and provides time-based information about performance. This VSM technique is based on generating a current state map that shows the current performance and conditions of the studied systems, and a future state map which serves as a target for

 ISBN: 978-1-61208-308-7

improvement actions. Its simplicity and effectiveness have led to VSM being effectively integrated into several applications, appropriate to both manufacturing and non-manufacturing situations. Although the lean concept originated in the automobile industry, the increased application of lean practices in healthcare has seen growth in the popularity of modeling tools such as VSM [9]. VSM has been successfully utilized as a lean implementation tool in many different healthcare systems, from small physician's clinics [10] to larger and more complex systems such as Emergency Departments.

Although VSM is very effective in presenting system parameters such as operation cycle times and resource capacities and availabilities, it does not have the ability to analyze the impact of system settings on performance. Various authors agree that the potential for lean healthcare exists, but evaluating its successful implementation remains a challenge [11]. Others argue that lean healthcare has too often been adopted unquestioningly and may actually result in more harm than good: [12] argue that the redesign of healthcare's complex processes 'leads to continued fragmentation of healthcare work, loss of autonomy for the health professions, and a potential increase in hospital misadventure'. This is due to the fact that VSM lacks prediction capabilities, so, it is also difficult to know if the desired level of system performance is the best that can be achieved. Moreover, value stream maps cannot take account of system variations and uncertainty [13], so, VSM must be integrated with another technique that can handle system variation, show dynamics between system components and validate the future state before any improvement steps can actually be implemented. Modeling and simulation can fulfill this need. Modeling and Simulation tools have the capabilities to fulfill this need.

Simulation can be used to master new business concepts such as agile and lean management [14]. Unlike VSM, simulation offers more thorough analysis of a system's data, including examining its variability, determining whether the data is homogenous, and estimating the probability distribution that fits the patterns of the data. This kind of in-depth analysis of data enables simulation to be used to support continuous improvement [15] and to model systems' future state maps, so, showing the ideal state of the system that can be pursued over time. The advantage of using the simulation approach in a lean context is not limited to the phase of developing a future state map, but extends to selecting the best alternative to the current system status.

III. PROJECT BACKGROUND

The tertiary partner hospital in this study is a private hospital in Dublin which provides a full range of services, including (among others) an Orthopedic centre, Oncology/radiotherapy care, eight operating theatres, and an Emergency Department (ED) that operates 12 hours a day, six days a week. There are two particular drivers for this project:

- Patient perceptions of Quality. Quality can have many definitions - a product's quality can refer to whether it works or not, looks good or not, adds value or not. In the case of service, quality is much more subtle - it depends on both the provider and the recipient - here, the patient– it is all about their perception of the experience. Delays in any process result in patients literally sitting or lying around. For a fully conscious patient this can be frustrating, as there is no perceived value in waiting, unless it is recovering after illness or surgery. However, many patients also attend the hospital as in-patients for diagnosis, so, there are plenty of opportunities and potential causes for delays.

- The continuous improvement ethos of the hospital. The hospital has been through the Joint Commission International Accreditation (JCI) process twice, which requires that all aspects of managing the hospital - from leadership to infection control to the patient journey - are clearly stated for all staff to see. This is to be achieved through written policies, procedures and guidelines, and the whole process of setting everything down in writing is a good way of spotting gaps in the service offered. The JCI process sees continuous quality improvement as a cornerstone to accreditation, and requires quality improvement be embedded in the organization's culture. The hospital has Key Performance Indicators (KPIs) and performance benchmarks. It constantly strives to improve its processes, and runs quality improvement initiatives, both large and small, both within departments and hospital wide.

The hospital's Quality Improvement Committee meets monthly to discuss policies, procedures and quality initiatives in the hospital. Amongst the quarterly statistics and KPIs it considers is the average Length of Stay (LOS) of patients. LOS figures are averaged over the whole hospital population, although in reality lengths vary according to patient diagnosis, and many factors contribute to LOS variations, both qualitative (individual doctor practice style, individual patient diagnosis) and quantitative (discharge policy implementation, bed supply, method of payment) [16]. Delays in patient discharge are due to various factors: inconsistent discharge rounds (doctors attending when they can), delays in waiting for tests and in discharge prescriptions and late referrals to allied health professionals. The delayed discharges have a negative impact on availability of acute beds, admissions of elective and emergency admissions and overall patient experience. Currently the discharge process in the partner hospital depends on when consultants conduct their ward rounds. If it is decided to discharge a patient it is recorded in the patient's chart. A junior doctor then organizes any medications and last minute tests, and the patient's discharge summary, and the nurse notes when the patient leaves in their electronic record on the Hospital Information System (HIS). Discharge data on the HIS shows discharges primarily occur after midday, which has a negative impact on the availability of beds for both elective and ED admissions. The admissions process for both elective and ED patients can be quite lengthy. The main problem is the assignment of a bed for the incoming patient, which can be affected in leaving patients are discharged late, which may not only lead to patients experiencing significant delays, but to them being assigned to an inappropriate ward if there is no bed available on the appropriate ward. This means consultants have to make

 ISBN: 978-1-61208-308-7

rounds to other wards as well as those they are primarily assigned to, and necessitates further ward transfers when an appropriate bed does become available. If the bed manager had better information regarding bed availability s/he would be able to make more efficient admission decisions. At the beginning of this study, there was too much waste in the system but where exactly it arises needs to be clarified. Operations meetings have also highlighted the need for the hospital to have strategies for coping with unexpected influxes of admissions, when bed availability becomes critical. This paper describes a collaborative project between the bed management team and the research team. The main objective is to analyze the situations which cause discharge delays and propose valid solutions.

IV. PROPOSED METHODOLOGY

As identified earlier, lean thinking helps remove or limit the impact of non-value adding steps in a process. However, lean management requires a team of people; it is an expensive method of process redesign. Implementing an unsuitable or ineffective change creates further waste, as well as frustration. Simulation modeling offers the opportunity to test process alternatives in a safe environment. The issue outlined (i.e., the admission-discharge process) is a day-to-day problem which has implications for strategic planning. As the work involves processes at the patient level Discrete-Event Simulation (DES) is suitable. The objective is to aid the bed administrator to allocate beds more efficiently, which will impact positively on the LOS of both the elective and ED admissions (both of which currently suffer delays due to the ad hoc discharge process) increasing patient satisfaction. An important added benefit is that this will help the hospital prepare better for any 'bed crises'. Using lean principles as the foundations, the framework devised for examining these issues involves 1) identification of the process; 2) identification of the value to the patients; 3) develop VSM for the process(es); 4) develop the simulation model; and 5) experimentation and analysis.

A. Identification of the Process

There are many distinct processes in a hospital, but they are all inter-connected. Fig. 1 presents the overall patient journey, whether they enter the process as an elective candidate or via the Emergency Department, where they will have been diagnosed as requiring further treatment as an in-patient. The Bed Manager then allocates the patient a bed, and they are given in-patient treatment until they are discharged home, or for further treatment elsewhere, or (in the worst case) to the mortuary.

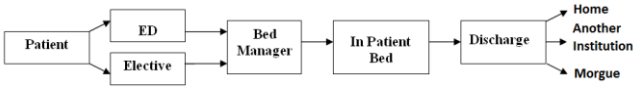


Figure 1. Overall admission-discharge process

B. Identifying the Value to the Patient

Literature shows a link between discharge and admission rates: both the elective and ED admission processes were reviewed separately with respect to the affect delayed discharge had on bed allocation. The time the patient spends in the admission process can be viewed as part of their overall LOS. Delays in either admission or discharge are non-value added 'activities' which directly affect the patient's perception of the quality of their diagnosis/treatment; delays for the patient are frustrating at best and life-threatening at worst. Other 'customers' who are indirectly affected by delays in the process include referring doctors (who want their patients to be treated effectively and efficiently) and insurance providers.

C. Develop Value Stream Maps for the process(es)

Value stream mapping (VSM) gives a pictorial representation of the flow of materials, people and process information from the start to the end of a process. It includes all activities involved in the process, whether they can be categorized as value-added (e.g., blood tests); non-value added necessary (e.g., the patient completing their insurance details); and non-value added unnecessary. This can highlight problems and can help identify their causes, assisting managers in prioritizing process improvements. The overall pictorial representation VSM gives can also enable other stakeholders (doctors, senior management and accreditation inspectors) to appreciate the process more easily and more fully. One the process to be mapped has been selected, VSM involves 1) talking to frontline staff involved in the process (here, bed manager, ED manager, admission clerks, etc.) to map each stage of the process on paper 2) collecting data to produce a current state map; and 3) conducting a critique of the current state to identify wasteful areas of the process which offer the best chance of being changed.

1) Elective Admission VSM

Fig. 2 shows the VSM of the admission of elective patients, with relevant information attached to each step of the process: 1) capacity (i.e., number of people available to perform that step); 2) type (i.e., personnel required to perform that activity); and 3) P/T: process time is the time required to complete the activity. The averaged time taken at each step of the process is recorded in a time line at the bottom of each image - for example, the time taken for the Bed Manager to assign a bed for an elective admission can take between 10 and 15 minutes, so, is averaged to 12 minutes. The timeline is presented from the patient's perspective, and includes process time and wait time. Process time represents value added activities, usually involving the patient (e.g., transfer to the patient room, attending radiology for a scan) while wait time is the time spent in an activity the patient is not involved (e.g., waiting for a bed to be assigned).

2) ED Admission VSM

The developed VSM constitutes of two parts – the main process (Fig. 3) and a sub-process (Fig. 4), which shows the patient having a scan in radiology. The associated time lines are shown at the bottom of each figure where these figures are the most likely value for the corresponding step (i.e. triangular distribution). The map was created to illustrate the impact delays in the diagnostic process can have on an ED patient's admission experience. Mapping the sub-process

 ISBN: 978-1-61208-308-7

adds complexity to the model, making it a truer representation of the real system and increasing the quality of the simulation model.

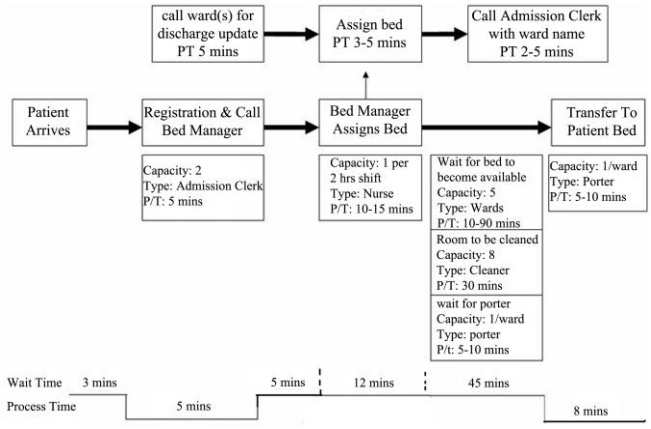


Figure 2. VSM Elective admission process

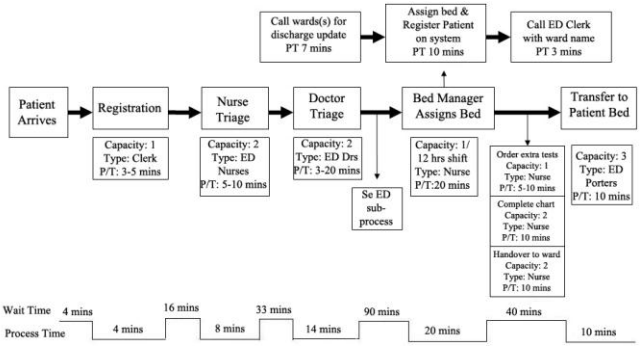


Figure 3. VSM ED admission process

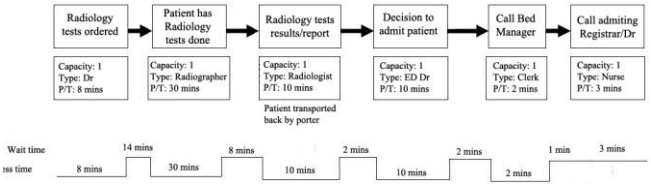


Figure 4. VSM ED admission sub- process

The times assigned to each step in the ED admission process was collected through observations carried by medical and administration teams. These times were summed according to type to determine where the overall process suffers most in terms of non-value added activities: the ED sub-process was evaluated separately as it contains both waiting and processing times. Table I shows the results.

TABLE I. VSM WAIT, PROCESS AND TOTAL TIMES

Time (mins)	Wait		Process	Total
Process	Non value add - necessary	Non value add - unnecessary		
Elective admission	3	62	13	78
ED admission	0	93	56	
ED sub-process	20	10	60	
	20	103	116	239

The total time taken for the elective admission is 78 minutes, which may seem surprising given that these patients are scheduled appointments. The total time taken for ED admissions (including the sub-process) is 239 minutes, nearly half of which this is non-value added elements composed of the waiting involved between steps. The two longest wait periods occurs when 1) diagnostic tests are required – these may involve long scan times (MRI) or preparation (patients drinking contrast medium before a CT scan). These activities can be seen as non-value-added but necessary in order to have the scan; and 2) Awaiting for a bed to be assigned and to become available, which are non-value added elements.

D. Developing the Simulation Model

The analysis of empirical data is essential in developing a robust simulation model that considers the time features of the examined system in terms of the volume and patterns of demands. Historical records were gathered from the hospital information system over a 6 months period, as provided by hospital managers. The data included occupancy rates, elective, ED and overall admission numbers. The average percentage of late discharges (after noon) was determined as 83 per cent (see Fig. 5).

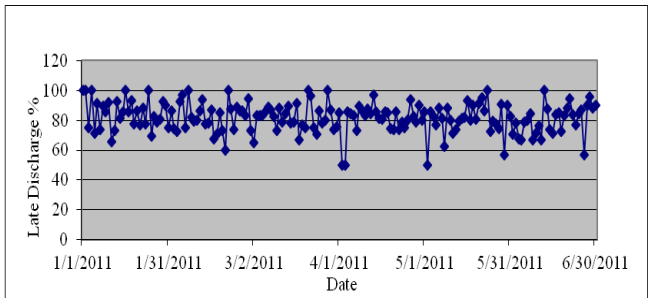


Figure 5. Late discharges as percentage of overall discharges

As the discharge process affects bed availability, average bed occupancy levels was determined to be 91% due to the high rate of late discharge (Fig. 6). Average bed occupancy levels were determined to be 91% (Fig. 6). We argue that the discharge process - and specifically, late discharges - is a significant factor in this figure, which we argue is too low. The simulation model was designed to reflect this relation between the late discharging rate and the bed management performance issues. The data collected was also analyzed to extract the arrival rates of patients after categorizing them to ED patients and out patients after they had been categorized into those needing admission, and those need out-patient care (Fig. 7).

Based on this analysis of empirical data and the VSM results, a comprehensive simulation model was developed for the admission-discharge cycle(s) in the hospital. Simulation model modules were connected to resemble the VSM processes, where blocks are connected to create conceptual flow chart, which simplifies the construction of the simulation model.

 ISBN: 978-1-61208-308-7

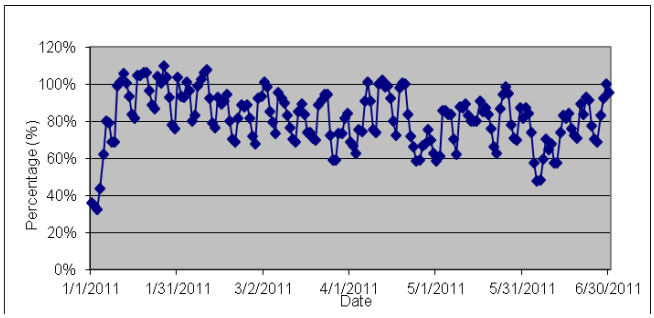


Figure 6. Bed occupancy levels over a 6-month period

Thus, the top level of the simulation model defines the overall model structure, with the sub-level blocks containing additional modules with more details. The simulation model was developed using ExtendSim package and object-oriented programming was used to customize pre-defined blocks. A database was used to save the measured KPIs (i.e., avg. occupancy levels, avg. LOS, and avg. waiting time for ED admission) after each simulation run, after which the simulation output was exported in tabular form for future analysis and validation.

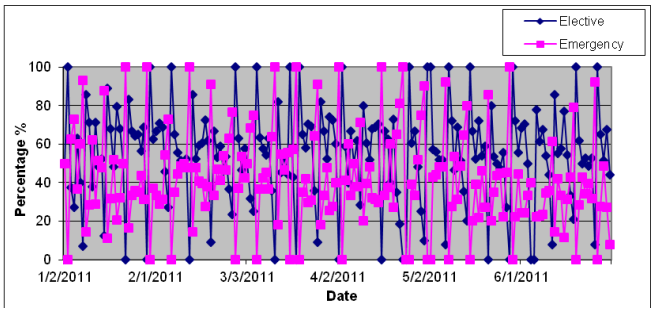


Figure 7. Number of admissions for ED and elective patients

E. *Verification and Validation*

Developing the simulation model depends on the right process being modeled and it being modeled (or built) correctly. As disconnects can exist between real world problems and models of those problems, the verification and validation processes are crucial. The model used six months data gathered from the HIS to predict average occupancy within four per cent of actual values. Literature states 10 per cent statistical accuracy as being an acceptable level (Connelly and Bair, 2004). Fig. 8 shows the correlation between predicted and actual occupancy rates.

Real (blue line) refers to the original data set, but these included bed in the observation ward, which opened on day 136. Their inclusion increased the number of total beds available, thus, skewing the data, so, it was decided to remove the observation beds from the analysis. This 'new' data is represented by the 'Real after modification' (green line). The 'Simulated' (red) line represents the values predicted by the model - as the figure shows, these two track each other.

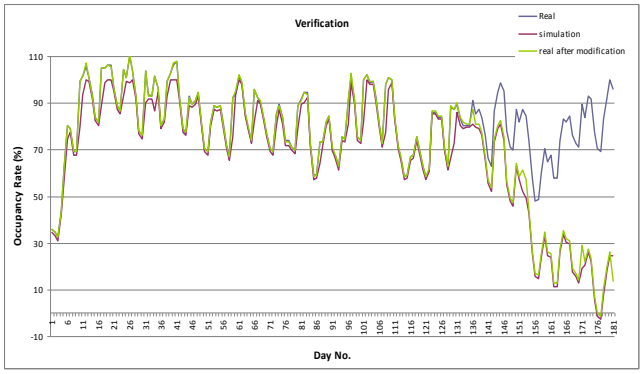


Figure 8. Correlation between predicted and actual values

V. EXPERIMENTATION AND ANALYSIS

A. *Scenario Design and analysis*

Bed availability becomes a significant problem if there is an increased demand due to ED admission rates or the effect of late discharges. Increasing the number of beds in the hospital and improving the discharge planning for patients were the main two solution options the management team wanted the model to examine and analyse. Two variables were introduced to examine these strategies: number of beds available and late discharge rate, and three main scenarios were then introduced, as shown in Table II.

TABLE II. SIMULATION VARIABLES FOR BASE SCENARIO AND SCENARIOS 1, 2, AND 3

	Number of beds	Late discharge rate
Baseline Scenario	145	83%
Scenario 1	110 to 170 with a step of 5	83%
Scenario 2	145	0 to 1 with a step of 0.1
Scenario 3	110 to 140 with a step of 5	0 to 1 with a step of 0.1

It could be expected that increasing the number of beds above the current level (i.e., 145 beds) would lower bed occupancy levels and decrease ED waiting times, while decreasing available bed numbers (without changing any other variables) would lead to significantly increased delays for patients waiting to be admitted (Table III). However, changing available bed numbers caused no significant change in penitents' average LOS in the hospital, because the discharge process still unchanged (i.e., the late discharge rate remained constant at 83%).

On the other hand the simulation shows that reducing the late discharge rate (i.e., scenario 2) would have a significant impact on the average LOS (see Table IV).

As the Table IV shows, continuously decreasing the proportion of late discharges constantly decreases bed occupancy levels, allowing the hospital the opportunity to decrease its number of available beds instead, with the attendant cost savings. To examine this possibility in more detail, the model examined (as scenario 3) a combination of

 ISBN: 978-1-61208-308-7

different discharge rates and the potential associated reductions in bed numbers.

TABLE III. SIMULATION RESULTS OF SCENARIO 1

Number of Beds	Bed Occupancy	ED Waiting Beds (hrs)	Avg LOS (days)
110	95	13.8	4.5
115	95	12.1	4.5
120	94	9.6	4.5
125	93	3.2	3.9
130	93	5.1	4.3
135	92	3.5	4.4
140	92	3.3	4.4
145	**91**	**2.4**	**4.4**
150	83	0.7	4.2
155	81	0.4	4.3
160	78	0.2	4.3
165	74	0.1	4.3
170	71	0.1	4.3

TABLE IV. SIMULATION RESULTS OF SCENARIO 2

Late Discharge Rate	Bed Occupancy	ED Waiting Beds (hrs)	Avg LOS (days)
0	51	0.0	2.1
10	56	0.0	2.4
20	61	0.0	2.7
30	64	0.0	2.8
40	70	0.0	3.2
50	73	0.0	3.4
60	79	0.2	3.7
70	85	0.9	4.1
80	**91**	**2.4**	**4.4**
90	93	6.7	5.0
100	98	13.7	5.8

Interestingly, as the rate of late discharges decreases, the possibility of reducing the number of beds increases, while the quality of care for patients in terms of waiting times for beds and LOS in hospital also improves (see Fig. 9). These results of the simulation scenarios suggest that increasing bed numbers can be considered a knee-jerk reaction and will only solve the problem temporarily; as once more beds become available the referral rate for elective procedures will also increase. From an operations perspective, new beds are very expensive because they have to be staffed appropriately and there may often be limited physical space available to respond to higher occupancy: so, this is not a realistic option.

It is clearly more efficient and cost-effective to review the discharge process to identify effective actions to decrease delayed discharges. This goal can be accomplished by early medical assessments (ward rounds), faster laboratory or diagnostic imaging results; fulfilling prescriptions in the pharmacy in a timely manner; and discharging patients to alternative care settings. The majority of patients' discharges are delayed because a nursing home bed, homecare packages, other community supports, rehabilitation facilities or other types of alternative care are not available. In an analysis of the reasons for the delay in discharging it was found that 75% of those patients were seeking nursing home care. Therefore providing short- and long-term beds has a substantial effect in reducing waiting times in many other stages of healthcare system. As acute hospitals become more technologically advanced in diagnostic and interventionist care, perhaps they are no longer appropriate settings for convalescing. Hospitals may want to consider 'step down units' as an option for the future, which would dove-tail well with the increasing current use 'hub and spoke' model. Hubs are based in large conurbations not served by tertiary care hospitals, and specialist clinics in these towns refer patients to the underlined studied Hospital for specific sub-specialty care such as orthopedic surgery, vascular surgery.

VI. LIMITATIONS AND FUTURE WORK

While the information gathered for the VSM came directly from people involved, the quality of the data was not actually audited, so, assumptions and estimates had to be used for some steps. The documentation methods used to record some steps of the processes – hand-written notes in the patient's physical chart and entries on the patient's electronic record - can make capturing data about those steps difficult. And while simulation modeling can present possible solutions and the impact changes can have on a process, it does not take account of all the varied and often challenging professional bureaucracy environments.

VII. CONCLUSION

Further investigation is also required to substantiate preliminary findings, as certain assumptions were made, primarily about the discharge process (as noted previously). So, future work should incorporate a small team of people directly involved in the processes to gain results data of richer quality. Use DES modeling in conjunction with system dynamics model to work with, rather than against, each other. Hospital processes are interdependent and their efficiencies (or wastes) affect others, so, a system-wide (aggregate) view should be modeled. The person most affected by the whole system is the patient. The DES can inform the SD model to identify potential synergies between processes. This work discusses the practicalities, challenges and limitations of applying lean healthcare modeling to a hospital process – so, it represents a risk assessment for process change. It incorporates the principles of lean management (focusing on the patient journey through a process and seeking to identify, and so reduce, unnecessary waiting and non-value-added activities) and simulation modeling (testing solutions to balance capacity and demand) to improve the delivery of healthcare to patients. The problem in question is the impact late discharge has on bed availability and admission processes.

Lean principles were applied to the problem, and the processes involved were mapped to achieve greater understanding and so inform the simulation model. Scenarios were run, focusing on the late discharge rate and its effect on numbers of bed required and patients' wait times overall LOS, allowing a future state map to be built. Simulation does not consider the environment or culture around a process - resistance to change may have to be addressed and countermeasures proposed - while a hospital may have a

 ISBN: 978-1-61208-308-7

change culture, this does not mean the consultants are parties to it.

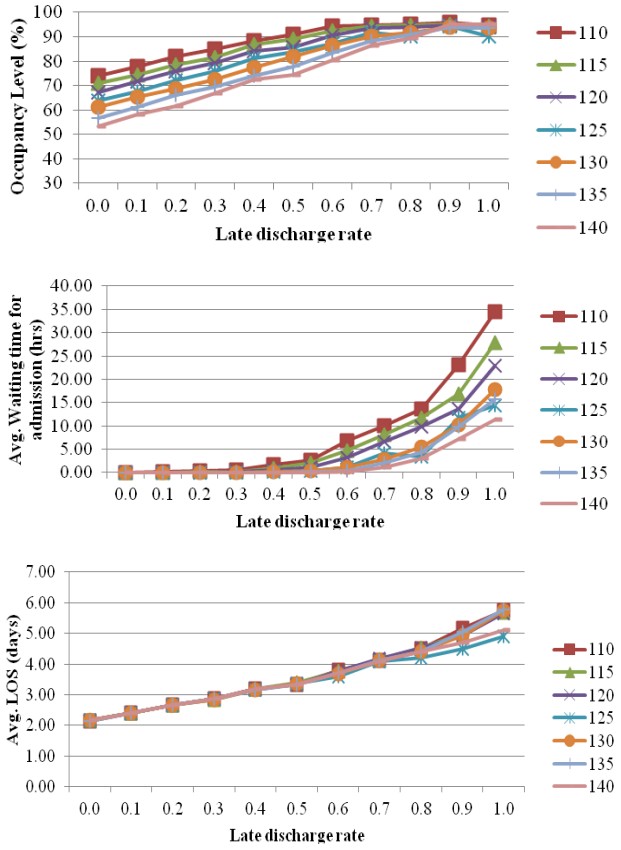


Figure 9. Simulation results of scenario 3

To achieve change, manager may have to seek opportunities for greater collaboration with consultants and include them in hospital change strategies.

REFERENCES

[1] S. B. Davis and P. J. Robinson, "Journal of Health Care Finance," vol. 37, no. 2, 2010, pp. 59–64.

[2] HIA, "The Private Health Insurance Market in Ireland," 2010. [Online]. Available: http://www.hia.ie/publication/consumer-surveys.htm. [retrieved: May, 2011].

[3] D. Martin and P. Singer, "A strategy to improve priority setting in health care institutions.," Health care analysis : HCA : journal of health philosophy and policy, vol. 11, no. 1, Mar. 2003, pp. 59–68.

[4] J. A. Bahensky, J. Roe, and R. Bolton, "Lean sigma--will it work for healthcare?," Journal of healthcare information management : JHIM, vol. 19, no. 1, Jan. 2005, pp. 39–44.

[5] M. J. Johnston, P. Samaranayake, A. Dadich, and J. A. Fitzgerald, "Modelling radiology department operation using discrete event simulation," in 18th World IMACS / MODSIM Congress, July. 2009, pp. 678–684.

[6] C. Jimmerson, D. Weber, and D. K. Sobek, "Reducing waste and errors: piloting lean principles at Intermountain Healthcare.," Joint Commission journal on quality and patient safety / Joint Commission Resources, vol. 31, no. 5, May 2005, pp. 249–57.

[7] M. Rother and J. Shook, Learning to see value stream mapping to create value and eliminate muda. The Lean Enterprise Inst., Brookline, Mass., 1998.

[8] R. K. Singh, S. Kumar, A. K. Choudhury, and M. K. Tiwari, "Lean tool selection in a die casting unit: a fuzzy-based decision support heuristic," International Journal of Production Research, vol. 44, no. 7, Apr. 2006, pp. 1399–1429.

[9] C. S. Kim, D. A. Spahlinger, J. M. Kin, and J. E. Billi, "Lean health care: what can hospitals learn from a world-class automaker?," Journal of hospital medicine : an official publication of the Society of Hospital Medicine, vol. 1, no. 3, May. 2006, pp. 191–9.

[10] R. R. Lummus, R. J. Vokurka, and B. Rodeghiero, "Improving Quality through Value Stream Mapping : A Case Study of a Physician ' s Clinic," Total Quality Management, vol. 17, no. 8, 2006, pp. 1063–1075.

[11] J. J. Waring and S. Bishop, "Lean healthcare: rhetoric, ritual and resistance.," Social science & medicine (1982), vol. 71, no. 7, Oct. 2010, pp. 1332–40.

[12] S. Winch and A. J. Henderson, "Making cars and making health care: a critical review.," The Medical journal of Australia, vol. 191, no. 1, Jul. 2009, pp. 28–9.

[13] C. R. Standridge and J. H. Marvel, "Why lean needs simulation" in Proceedings of the 2006 Winter Simulation Conference L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds., 2006, pp. 1907–1913.

[14] D. M. Ferrin, M. J. Miller, and D. Muthler, "Lean sigma and simulation, so what's the correlation? V2," in Proceedings of the 2005 Winter Simulation Conference M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds., 2005, pp. 2011–2015.

[15] M. Adams, P. Componation, H. Czarnecki, and B. J. Schroer, "Simulation as a tool for continuous process improvement," in Proceedings of the 1999 Winter Simulation Conference P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, eds, 1999, no. Grief 1991, pp. 766–773.

[16] A. Clarke, "Why are we trying to reduce length of stay? Evaluation of the costs and benefits of reducing time in hospital must start from the objectives that govern the change," Quality in Health Care, vol. 5, 1996, pp. 172–179.

 ISBN: 978-1-61208-308-7

Simulation Model of a Bus Line in Changing Traffic Conditions

Marek Bauer
Cracow University of Technology
Chair of Transport Systems
Cracow, Poland
mbauer@pk.edu.pl

***Abstract*—This paper presents the author's stochastic model of bus line operation, taking into account the variability of traffic conditions, as a result of disturbing factors. General structure of the model bases on graph and events theory. In this approach, bus line has been described by bus stops and sections between these stops, and the most essential traffic processes: running following sections, alighting and boarding passengers at bus stops, waiting for the possibility to departure. In the mathematical description of the simulation model, a dynamical system of matrix-vector equations is used. The fundamental element of presented model is the state vector of the system. It includes all important information about operation of each bus line: variables allowing to define location of each bus on a line in time and space, variables describing the occupancy of all buses and values of scheduled ride times which enable to evaluate punctuality. All parameters of the model have been determined with statistical and stochastic methods using, on the basis of extensive and identification own research including various types of sections (with or without separated bus lanes) and diversified location and types of bus stops. Presented simulation model can find wide application among others: to create feasibility studies for investments in public transport, evaluation of solutions streamlining bus traffic, planning bus routes and developing timetables.**

Keywords- urban transport; bus lane; simulation

I. INTRODUCTION

Operation of public transport requires permanent quality control of provided services. It is only with complete knowledge about the state of public transport, when efficient continuous improvement activities can be carried out. For this reason, public transport quality research should be conducted in a manner possibly continuous, in order to enable evaluation of not only current quality indicators but also efficiency of measures taken, such as: separated bus lanes and priorities in traffic lights.

Bus service is far more prone to traffic disorders than rail transport due to moving, mostly, in the stream of other vehicles. Currently, in Polish cities, one observe the increasing influence of overcrowded streets on deterioration of bus traffic conditions leading even to total breakdown of punctuality and regularity. The speeds achieved by buses vary, even within the same traffic route and the same time period. There are many factors influencing travel times of buses (e.g., [1], [2]), like the infrastructure of streets, intersections, and bus stops, the traffic conditions and organization of traffic, as well as the motoric and behavioral conditions.

General rules of bus public transport modeling and optimization were shown in publications [1], [3] and [4]. There are many publications with micro-simulation point of view. In [5], Mahmoud and Hine described a multi-criteria evaluation of user perception towards bus services and measures the gap in the perceptions held by current and potential users. Paper [6] illustrates a new method of calibration of bus performance parameters in the microscopic scale. Liu and Sinha [7] considered the problem of reliability of an urban bus network using a dynamic micro-simulation model framework. All above approaches are characterized by high level of accuracy.

However, micro-simulation models are very problematic due to the large number of inputs. For example – in Polish conditions, measurements in public transport consist of occupancies and stop-to-stop travel times registration. However, traffic volume measurements are usually not conducted at the same time. The range and the quality of the available information are usually not sufficient to be used in micro scale models for large areas of the city. On the other hand, the macro-simulation models have too general character. However, there is a gap between the highly accurate micro-simulation models and models in macro scale. Therefore, it was decided to build a model in which the reference point will be the efficiency of a bus line. This model should take into account traffic condition and various kinds of street infrastructure.

The aim of the author's own research is among others to determine:

- What is the influence of disturbing factors onto bus line operation?
- What functional parameters (including: average travel and running speeds and their variation) could be obtained in predetermined conditions under the influence of different external factors?
- What is the efficiency of separated bus lanes?

Answering these questions at the stage of investment planning is the basis of adequate implementation of privileges for bus public transport. A tool facilitating evaluation of potential investment effects presents a

 ISBN: 978-1-61208-308-7

simulation model of a bus line operation in differentiated traffic conditions. This model could be applied to the analysis of each bus line or public transport corridor, operating in urban conditions, taking into consideration streets' infrastructure and traffic conditions. The model can be helpful for time-table's better designing, in scheduling procedures, in planning and designing bus routes, and in network analysis (with the four steps approach [8]), in estimating input data for macro-simulation models.

In section 2, the general structure of the model is described, whilst in Section 3, the mathematical model based on the system matrix equations was shown. Section 4 explains two important models' elements: bus running time between following stops and initial conditions for every course on the bus line. In Section 5, an example of possible simulation results is presented. Finally, Section 6 presents the conclusion of the paper.

II. GENERAL STRUCTURE OF THE MODEL

It has been initially assumed, that the model will reflect processes occurring on a bus line, on meso scale. It will help obtain output data more precise than in the case of commonly applied macro-simulation models. At the same time, the model does not require such a great volume of input data as in micro-simulation models. Previously, this kind of approach was used in [9], where mathematical description of an urban bus route in peak hour traffic was presented.

The structure of the bus line operation model is based on graphs and events theories. The graph theory gives a possibility of selecting basic elements of any bus line. Whereas, the theory of events is very useful in description of all the special events taking place during bus line operation.

According to the graph theory, any bus line can be described as a simple digraph structure (Figure 1), where the set of vertexes consists of bus stops on the line, and set of edges consists of sections between these stops.

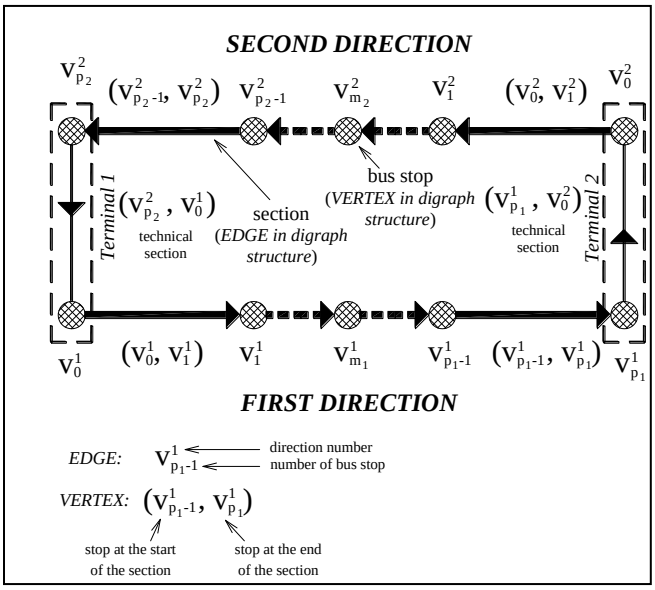


Figure 1. General structure of the model of bus line operation.

Majority of bus lines in European cities (excluding circle lines) operates in two directions. Last stop in first direction is situated on the same terminal as the first stop of second direction. Sections between these stops have technical character, and are also taken into consideration.

The description of model's elements is shown in Table I.

TABLE I. SETS OF ELEMENTS IN DIGRAPH STRUCTURE

Name of set	Description
Set of bus stops	$V^P(D) = \{v_m; m = 0,..., p\}$
Set of stop-to-stop sections	$A^O(D) = \{(v_m, v_{m+1}); v_m \in V^P(D); m = 0,..., p-1\}$
Set of technical sections	$A^T(D) = \{(v_p, v_0);\ v_0, v_p \in V^P(D)\}$

In this moment, the bus line can be treated as completion of individual modules: "between stops section" – "stop", located at the end of this section. This approach could be used in stop-to-stop travel time analysis. On the basis of times of departure from following stops, one is able to establish the time span between individual stops, forming the basis for constructing schedules and also the values of stop-to-stop speeds. As well, punctuality and regularity indicators can be calculated. It is however not possible, to indicate reasons for current and potential disorders on the bus line, as the information about the time between departures does not give any insight into the structure of travel time between following stops – how much time of it is the running time and how much is dwell time. As a result, it is difficult to estimate, if extended time of module completion on a line is caused by difficult traffic conditions on a section, or if it is the effect of exceeding traffic capacity of a stop, or increased numbers of alighting and boarding passengers. In the same way, positive effects of privileges for buses (e.g., separated bus lanes) are not easy to extract.

For describing the traffic processes occurring on the bus line, the methodology of discrete events has been applied. The essence of which is the assumption that times of subsequent events are predictable on the basis of previous events occurrence. The simulator does not take into account the state of a model between subsequent events, it reacts to specific events occurring subsequently and the model status remains unchanged – until it changes, as a result of previous event. Simulation watch moves until the next event from the events' list occurs and then operations provided for this event are performed.

On every stop-to-stop section, a large group of possible events influence onto travel time. They are connected with any possible stoppings during the line operation at the stops areas and at the intersections, e.g., the: moment of stopping at the inlet of intersection (because of red signal), the moment of stopping just before the stop (because of busy stop positions), the moments of start and end of alighting and boarding passengers, the moments of start and end of opening door, the moment of physical departure from the stop. They have fundamental influence on the state of the bus line, but their number is individual for any course on the line.

Only few of them are obligatory, while the rest might occur or not.

In the presented simulation model, it is assumed that only the most important discrete events should be taken into consideration (Figure 2). Three possible events have strong influence onto current state of the line: moment of start alighting and boarding passengers, moment of end alighting and boarding and moment of departure from stop.

These three discrete events always have a place (except request stops), and start three of the most significant processes on every stop-to-stop section:

- **Running time** – defined as time interval between the moment the bus is starting from one stop and stopping at the next one.
- **Alighting and boarding time** – defined as time period between the moment of start and the moment of end of alighting and boarding passengers, even if door is still open.
- **Time of waiting for possibility to departure** – defined as time between the moment of end of alighting and boarding passengers to the moment of physical departure from the stop.

According to Figure 2, the stopping time consists of alighting and boarding time and time of waiting for possibility to departure, because of impossibility of start its movement.

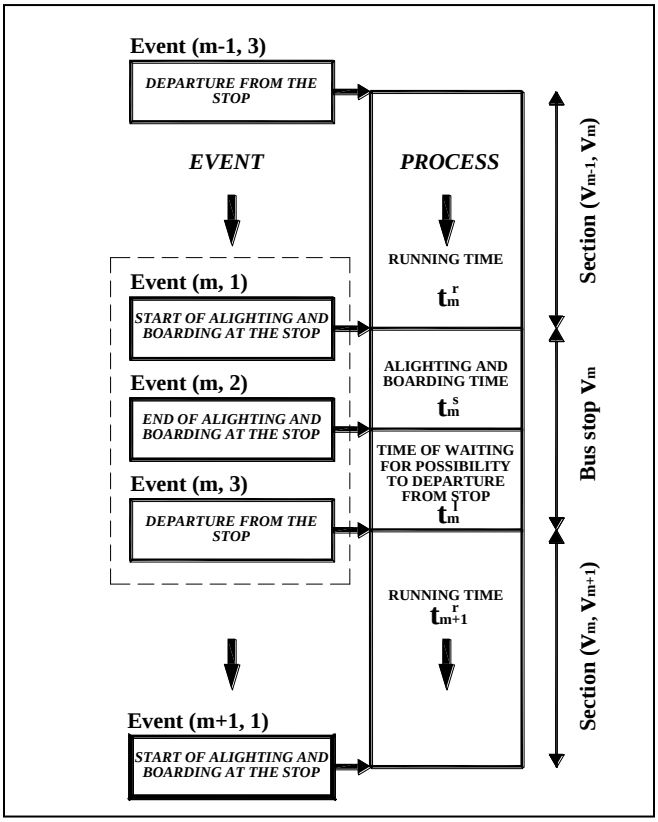


Figure 2. Possible events during one stop-to-stop operation.

All remaining events were aggregated to these three processes. The biggest simplification concerns running time, where the number of possible influences is the largest. But, because of the planning character of the model, and its usefulness – this simplification was assumed. If necessary, this approach can be completed, as in [10].

On the first and last stop (on terminals) – only one event is important. In case of beginning stop (number "0"), only the moment of departure is important (event 3), in case of the last stop on the line – only the moment of start of alighting and boarding is considered (event 1). Moment of departure from first stop can be estimated as semi-random variable.

The last part of the structure of the model – is a group of buses operating on the line. Final description on every event on the bus line has the form (*k*, *j*, *m*, *i*), where:

- k – course number (1, 2, 3, …; even numbers – first direction, odd numbers – second direction),
- j – individual number of the bus on the line (j = 1, 2, …),
- m – number of stop (m = 0, 1, 2, …, p),
- i – number of event (i = 1, 2, 3).

Set (*2*, *3*, *10*, *3*) means that bus number 3 during course number 2 has just departed from the stop number 10.

Was also assumed, that model should have stochastic character. Full description of the model's structure is presented in [11].

III. THE MATRIX MODEL

For reasons of utility of presented model, it is critical to capture dynamics of individual processes occurring on a line, in particular the changes of location of subsequent buses in time and space. Vehicle movement and connected with it, the passenger relocations (described with events), result in changes of the system state, the system being a bus line. Earlier described elementary occurrences happening during each module (between stops section – bus stop) are the grounds for developing a matrix system.

Knowledge of the system state in a current moment and of the past system inputs (especially those of random character), allows to specify the system state at later moments. Dynamics of the model is taken into account by changes of traffic conditions on sections between bus stops, random changeability of passengers' streams at bus stops and thanks to including the influence of time the buses spend at the final terminals on starting moments, and also duration of subsequent courses.

A. Matrix equations

The proposed model of a bus line operation can be expressed as dynamic vector and matrix system of discrete events, as following:

$$x_{k,j,m+1,1} = A^r \cdot x_{k,j,m,3} + B_1^r \cdot u_{k,j,m+1}^{rw} + B_2^r \cdot u_{k,j,m+1}^{rn} + C^r \cdot w_{k,j,m+1}^r \quad (1)$$

$$y_{k,j,m+1,1} = D^r \cdot x_{k,j,m+1,1} + E_1^r \cdot x_{k,j,m,3} + E_2^r \cdot x_{k-n,j-1,m+1,1} + E_3^r \cdot x_{k,j,0,3} \quad (2)$$

$$x_{k,j,m+1,2} = A_1^s \cdot (x_{k,j,m+1,1})^2 + A_2^s \cdot x_{k,j,m+1,1} + B^s \cdot u_{k,j,m+1}^s + C^s \cdot w_{k,j,m+1}^s + v_{k,j,m+1}^s \quad (3)$$

$$y_{k,j,m+1,2} = D^s \cdot x_{k,j,m+1,2} + E_1^s \cdot x_{k,j,m+1,1} + E_2^s \cdot x_{k,j,0,3} \quad (4)$$

$$x_{k,j,m+1,3} = A^l \cdot x_{k,j,m+1,2} + B^l \cdot u_{k,j,m+1}^l + C^l \cdot w_{k,j,m+1}^l \quad (5)$$

$$y_{k,j,m+1,3} = D^l \cdot x_{k,j,m+1,3} + E_1^l \cdot x_{k,j,m+1,2} + E_2^l \cdot x_{k-n,j,m+1,3} + E_3^l \cdot x_{k,j,0,3} \quad (6)$$

The occurrence of each event on a line (taking into consideration every bus operating on the bus line) is defined by a system of two equations:

- equation of the system state (i.e., (1), (3), (5)),
- equation of system outputs (i.e., (2), (4) and (6)).

The description of all model vectors is given below.

B. Vectors

The most vital element of the system is the state vector defined only at discrete moments of time, responding to occurrence of subsequent events on a line (Table II). It consists of current time of simulation (for every vehicle on the line in every determined moment of time), occupancy of vehicle, current distance from starting stop (every vehicle on the line) and scheduled, cumulated travel time – needed for punctuality indicators.

Vectors of deterministic system inputs are assigned to basic movement processes occurring on a line. Two of them include information influencing travel of the section between bus stops, with and without bus lane – these are section lengths and the number of crossroads located on the sections. The vector of deterministic inputs influencing the passenger exchange includes average intensities of alighting and boarding passenger streams, while vector of inputs connected with bus waiting for the possibility to leave the bus stop focuses average standard time of waiting, average length of extending the waiting time and time of covering a section according to a time-table.

Vectors of random system inputs represent uncertainty at the system input and contain random component of the time of covering the section between bus stops, of passengers exchanges and of waiting for the possibility to leave the bus stop, as well as random components of the number of passengers alighting and boarding at the bus stops.

Random components for the duration of processes are established as random variables from normal distribution of zero average values and variances defined for particular types, respectively, buses, stops and between stops sections. The random components of the number of passengers alighting and boarding have been modeled as random variables established from the Poisson distribution.

TABLE II. INPUT VECTORS IN PRESENTED SIMULATION MODEL OF BUS LINE OPERATION

Name of the vector	Vector	Description
State vector	$\mathbf{x}_{k,j,m,i} = \begin{bmatrix} T_{k,j,m,i} \\ P_{k,j,m,i} \\ L_{k,j,m,i} \\ R_{k,j,m,i} \end{bmatrix}$	Current simulation time Current number of passengers inside vehicle Current length Schedulet travel time
Vector of deterministic system inputs (section with separated bus lane)	$u_{k,j,m+1}^{rw} = \begin{bmatrix} s_{m+1}^{rw} \\ 0 \\ l_{m+1}^{rw} \\ 0 \end{bmatrix}$	Number of signalized intersections - Length of section with separated bus lane -
Vector of deterministic system inputs (section without bus lane)	$u_{k,j,m+1}^{rn} = \begin{bmatrix} s_{m+1}^{rn} \\ z_{m+1}^{rn} \\ l_{m+1}^{rn} \\ 0 \end{bmatrix}$	Number of signalized intersections Number of non-signalized intersections Length of section without bus lane -
Vector of deterministic system inputs (alighting and boarding)	$u_{k,j,m+1}^{s} = \begin{bmatrix} \lambda_{m+1}^{a} \\ \lambda_{m+1}^{b} \\ 0 \\ 0 \end{bmatrix}$	Average intensity of alighting passengers Average intensity of boarding passengers - -
Vector of deterministic system inputs (waiting for possibility to departure)	$u_{k,j,m+1}^{l} = \begin{bmatrix} \bar{t}_{k,j,m+1}^{lz,z^P} \\ \bar{t}_{k,j,m+1}^{lw,z^P} \\ 0 \\ r_{k,j,m+1} \end{bmatrix}$	Average normal time lost by bus at the stop Average extended time lost by bus at the stop - Scheduled stop-to-stop running time
Vector of random system inputs (section)	$w_{k,j,m+1}^{r} = \begin{bmatrix} \hat{t}_{k,j,m+1}^{rw,z^O} \\ \hat{t}_{k,j,m+1}^{rn,z^O} \\ 0 \\ 0 \end{bmatrix}$	Random component of running time (bus lane) Random component of running time (common lane) - -
Vector of random system inputs (alighting and boarding)	$w_{k,j,m+1}^{s} = \begin{bmatrix} \hat{t}_{k,j,m+1}^{s,z^A} \\ \hat{a}_{m+1} \\ \hat{b}_{m+1} \\ 0 \end{bmatrix}$	Random component of alighting and boarding time Random component of the number alighting passengers Random component of the number boarding passengers -
Vector of random system inputs (waiting for possibility to departure)	$w_{k,j,m+1}^{l} = \begin{bmatrix} \hat{t}_{k,j,m+1}^{lz,z^P} \\ \hat{t}_{k,j,m+1}^{lw,z^P} \\ 0 \\ 0 \end{bmatrix}$	Random component of normal time lost at the stop Random component of extended time lost at the stop - -
Vector of constant system inputs	$v_{k,j,m+1}^{s} = \begin{bmatrix} v_{k,j}^{s,z^A} \\ 0 \\ 0 \\ 0 \end{bmatrix}$	Constant value - - -

System outputs are tightly linked to the system state. There can be distinguished 11 output values, which are included in three vectors (Table III). These are: drive duration at the moment of beginning passenger exchange, actual time of covering a section, current distance from the starting stop, interval between vehicles at the time of reaching a bus stop, drive duration at the moment of finishing passenger exchange, alighting and boarding time, number of passengers on a bus leaving a bus stop, drive duration at the moment of leaving the bus stop, time of waiting for the possibility to leave the bus stop, deviation from the time table, interval between vehicles at the moment of vehicle leaving the stop. On the basis of system outputs line functioning quality indicators can be determined, including indicators of punctuality and regularity and travel comfort.

TABLE III. OUTPUT VECTORS IN PRESENTED SIMULATION MODEL OF BUS LINE OPERATION

Name of the vector	Vector	Description
Output vector (between stops running)	$y^r_{k,j,m+1} = \begin{bmatrix} T^r_{k,j,m+1} \\ t^r_{k,j,m+1} \\ L_{k,j,m+1} \\ h^r_{k,j,m+1} \end{bmatrix}$	Travel time from beginning stop
		Section running time
		Distance from beginning stop
		Between buses interval (before stopping)
Output vector (alighting and boarding)	$y^s_{k,j,m+1} = \begin{bmatrix} T^s_{k,j,m+1} \\ t^s_{k,j,m+1} \\ P_{k,j,m+1} \end{bmatrix}$	Travel time from beginning stop
		Alighting and boarding time
		Occupancy of vehicle
Output vector (waiting for possibility to departure)	$y^l_{k,j,m+1} = \begin{bmatrix} T^l_{k,j,m+1} \\ t^l_{k,j,m+1} \\ d_{k,j,m+1} \\ h_{k,j,m+1} \end{bmatrix}$	Travel time from beginning stop
		Time of waiting for possibility to departure
		Deviation from time table
		Between buses interval (after stopping)

The remaining elements of the model (described in [11]), are: matrixes of status, inputs and outputs containing among others, established on the basis of empirical studies, running time parameters, dwell time and time of waiting for the possibility to leave a bus stop.

IV. ESTIMATION OF CHOSEN MODEL COMPONENTS

All matrix elements have been estimated with the help of statistical methods, on the grounds of vast measurements' results of bus operation in four Polish cities – mostly with the use of GPS receivers. A similar method was used in [12]. In total, more than 21 000 departures from the bus stops have been registered. Below, the results of estimation of selected model's parameters are presented.

A. Running time

The most significant element of a single module on the bus line is the running time. At the same time it is the most difficult element to determine, due to the number and influence of distracting factors. Even in very similar traffic conditions, it is possible to obtain very different travel times of subsequent buses of a line. In particular, substantial differences of travel times occur at the sections where buses travel on general access lanes, which are typical for high use of capacity and even exhausting the capacity. In the model, diversity of sections between stops has been taken into account – firstly – in terms of existing or not separate bus lanes, next, in terms of the way to isolating or the degree of using conventional capacity of a lane.

In case of section with separated bus lanes, average running time for individual types of sections between bus stops has been modeled with linear function of multiple regression:

$$\bar{t}^r_m = \beta^r_l \cdot l_m + \beta^r_s \cdot s_m \quad (7)$$

Dependent variables in this model are: the length of the section between following stops (l_m [km]) and the number of intersections with traffic lights located on this section (s_m [-]).

Sections with separated bus lanes differ among each other, therefore, they have been diversified in terms of number of cars turning right with bus lane using, and also number of maneuvers connected with access to targets, located by the lane, including parking lots on pavements.

Regression formula in case of section with separated bus lane, without essential influence of turning right vehicles' traffic (type AB) – on the basis of measurements: less than 100 [veh/h]) – on the inlet of signalized intersection is as follow:

$$\bar{t}^{r,AB}_{m+1} = 1{,}14 \cdot l_m + 0{,}38 \cdot s_m \quad (8)$$

If, on the section is located at least junction on which more than 100 [veh/h] turning right with bus lane using (type AS), then, the average running time can be estimated from the other formula:

$$\bar{t}^{r,AS}_{m+1} = 2{,}17 \cdot l_m + 0{,}27 \cdot s_m \quad (9)$$

Whereas, in case of frequent (evaluated subjectively) additional maneuvers on the bus lane (type AP), connected with access to targets, located by the lane (e.g., parking, supplies), average running time can be calculated by the formula:

$$\bar{t}^{r,AP}_{m+1} = 2{,}87 \cdot l_m + 0{,}18 \cdot s_m \quad (10)$$

Bus running time on the sections without separated bus lanes can be described by similar regression formula with one additional variable. It is number of intersections z_m [-], without traffic lights, where buses perform subordinate relations:

 ISBN: 978-1-61208-308-7

$$\bar{t}_m^r = \beta_l^r \cdot l_m + \beta_s^r \cdot s_m + \beta_z^r \cdot z_m \quad (11)$$

Sections without separated bus lanes have been classified in terms of their traffic conditions, assessed by planning method. They are determined on the basis of the traffic ratio of measured hourly traffic volume to estimated value of critical, planning volume, for one direction, *r* [-], described in [13]. These critical planning volumes refer to situations in which the car drivers looking for alternative travel paths. The traffic volumes should come from measurements – or in case of new investments – from microscopic analysis. Below, individual formulas for all specified cases were shown.

As very good traffic conditions (defined as type ZA), there were evaluated cases, when traffic ratio *r* is smaller than 0,5. In this situation, the bus speeds are high and limited only by drivers:

$$\bar{t}_m^{r,ZA} = 1{,}77 \cdot l_m + 0{,}28 \cdot (s_m + z_m) \quad (12)$$

In this case, running times can be shorter even than in case of sections with separated bus lanes (type AB).

Sections with good traffic conditions were defined as sections, where the traffic ratio is smaller than 0,8 (type ZB). In practice, occasional blockings of the buses take place:

$$\bar{t}_m^{r,ZB} = 2{,}72 \cdot l_m + 0{,}16 \cdot (s_m + z_m) \quad (13)$$

Medium traffic conditions are most commonly found in big Polish cities (type ZC). In this case (no congestion), bus speeds are strongly limited by other lane users. This time, the traffic ratio is smaller than 1,0.

$$\bar{t}_m^{r,ZC} = 4{,}08 \cdot l_m + 0{,}13 \cdot (s_m + z_m) \quad (14)$$

In presented model, the congestion (type ZD) was defined as poor traffic conditions, where traffic ratio is bigger than 1,0, but smaller than 1,2. In this case, many car drivers look for alternative paths, while buses move slowly with frequent stoppings – not only on the inlets of intersections. Therefore, the number of junctions is not relevant. Regression formula has the form:

$$\bar{t}_m^{r,ZD} = 6{,}47 \cdot l_m \quad (15)$$

The last type of section (ZE) corresponds to a critical traffic conditions, where buses operate in permanent congestion – the traffic ratio is higher than 1,2:

$$\bar{t}_m^{r,ZE} = 8{,}99 \cdot l_m \quad (16)$$

Random components of travel times on sections between bus stops are generated as random variables from Normal distribution with zero mean and variance defined with the function of section length.

B. *Initial conditions*

It is of great importance to establish initial values for simulation of each course on the line, beginning at the first stop (at the terminal). In the model, current time of every course duration depends on the average value and random component of deviation from time table on the last stop, during previous course completed on this terminal. This problem was also considered in [14].

Average value of deviation on the initial stop is determined depending on the length of actual reserve of operational time on the terminal or the volume of delay of reaching the terminal in relation to the time-table moment of scheduled departure during the next course. The value of the deviation random component is estimated with the use of Normal distribution with zero mean and standard deviation estimated on the basis of empirical studies. In order to avoid extreme cases, during simulation the "three sigma" rule has been included.

Occupancy of a vehicle leaving the initial stop equals the number of passengers boarding at this stop. It results from the assumption that the bus approaching the bus stop is empty and it is only possible for the passengers to board. The number of passengers boarding at the initial stop is estimated as a random variable from Poisson distribution, where the parameter is the average value of passengers' stream intensity and the length of current (simulated) line interval. So, on the first stop on the line, state vector can be thus represented in the form:

$$x_{k,j,0,3} = \begin{bmatrix} R_{k,j,0} - d_{k,j,0} \\ b_{k,j,0} \\ 0 \\ R_{k,j,0} \end{bmatrix}$$

Other model's elements have been described in [11]. For example, the average alighting and boarding time were modeled by non-linear multiple regression formulas, as the dependence from numbers of alighting and boarding passengers and current occupancy of the bus – for 6 types of vehicles (midi, normal and maxi – with low and high level of floor). Average time of waiting for possibility to departure was varied depending on the bus stop location (near side, far side nearest intersection, on the section) and location of stop positions on the lane or on bay.

V. SIMULATION RESULTS

Numerical execution of the model represents author's simulation software, called "AUTOBUS" – developed in the Mathematica 6.0 environment. It has been created on the basis of demonstrated mathematical, stochastic model. The simulation model has been verified by comparing the results from simulations of two urban lines with the results of independent measurements.

Functioning of the simulation model has been presented on the example of existing Cracow's bus line No. 130. The line was modeled by taking into consideration real traffic conditions and true values of passengers' streams. Calculations were conducted for afternoon peak hour, in

 ISBN: 978-1-61208-308-7

order to obtain conclusive results of such a simulation – 20 hours were executed and compared. As a result, the set of all variables defined in output vectors was obtained. Additionally, "AUTOBUS" offers set of statistics, for example: confidence intervals for all calculated variables. Figure 3 shows the travel time on the line 130 (only one direction), while Figure 4 shows the deviations from time table.

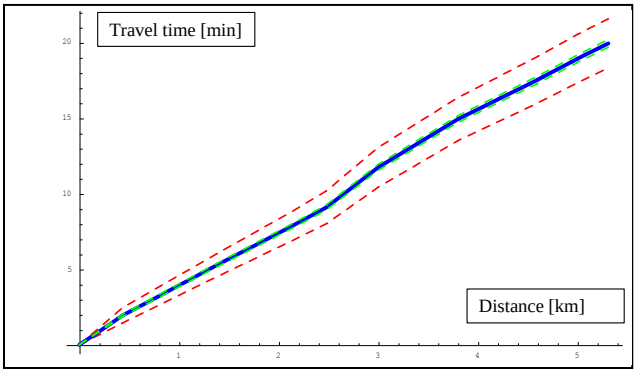


Figure 3. Example of simulation results –travel time on bus line 130 (blue: mean value of travel time, green – confidence interval for average travel time, red – shortest and longest travel time).

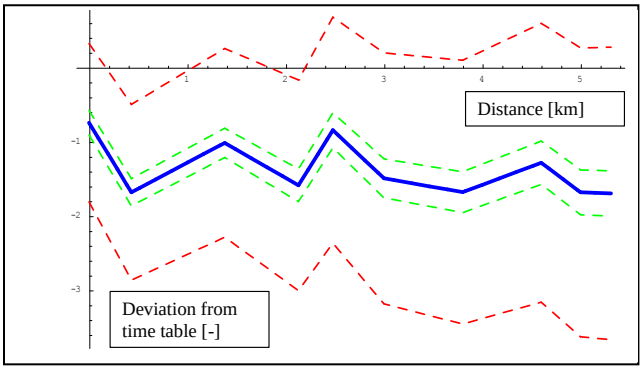


Figure 4. Example of simulation results –deviations from time table on bus line 130 (blue: mean value of deviation, green – confidence interval for average deviation, red – slowest and highest deviation from time table).

Similarly, in this way, the changing passengers' streams can be shown, separately for every bus and for whole bus line.

VI. CONCLUSIONS

A realization of the bus line operations is assumed to be a sequence of running times between following stops and times spent by buses at the stops. In this paper, a single bus line model in meso scale was presented, which could help to close the gap between existing micro and macro simulation models. This model will make easier the comprehensive analysis of any bus line operating in urban area. It could be used in:

- Scheduling procedures, especially in case of new lines, as a first approach, before starting and during first phase of operation (designed schedules have to be verified by measurements).
- Network analysis of public transport, in estimating input data for macrosimulation models of bus networks (e.g., VISUM software). Model could be useful also for better network calibration.
- Feasibility studies of new network elements, when a more detailed approach is not required.

In the further approach, additional elements will be attached to the model. For better stopping time description also the moments of stoppings just before the stop will be considered. The next step should be extending the model of effective priorities in traffic lights. Till this moment, in Polish cities, this kind of improvements for buses was implemented very seldom.

REFERENCES

[1] Transit Cooperative Research Program TCRP, Report 100, Sponsored by the Federal Transit Administration, "Transit Capacity and Quality of Service Manual," 2nd Edition, Transportation Research Board, 2003.

[2] M. Bauer, "Analysis of bus lanes' incomplete functional efficiency, (in Polish)" VIII Conference Transportation Problems in Cities in Motorized Congestion, P15-17 June 2011, Poznań-Rosnówko, Poland.

[3] V. R. Vuchic, "Urban Transit. Operations, Planning and Economics," John Wiley & Sons, Inc., Hoboken, New Jersey 2005.

[4] A. Ceder, "Public Transit Planning and Operation: Theory, Modeling and Practice," Elsevier, Butterworth-Heinemann, Oxford, UK, 2007.

[5] M. Mahmoud and J. Hine "Using AHP to measure the perception gap between current and potential users of bus services," Transportation Planning and Technology, Vol. 36, No. 1, pp. 4-23.

[6] J. Zhang, N. Hounsell and B. Shrestha, "Calibration of bus parameters in microsimulation traffic modelling," Transportation Planning and Technology, 2012, Vol. 35, No.1, pp. 107-120.

[7] R. Liu and S. Sinha, "Modelling Urban Bus Service and Passenger Reliability," The Third International Symposium on Transportation Network Reliability, 19-20 July 2007, The Hague, Netherlands.

[8] A. Szarata, Simulation analysis of CO_2 emission for different land use development schemes, Archives of Transport, Vol. 24, Issue 4, Warsaw 2012, pp. 579-591.

[9] P. Anderson and G. Scalia-Tomba, "A mathematical model of an urban bus route," Transportation Research, 4B, 1981, pp. 249-266.

[10] R. Bąk, "Simulation Model of the Bus Stop," Archives of Transport, Vol. 22, No 1 / 2010, pp. 6-25.

[11] M. Bauer, "Influence of street's infrastructure onto bus public transport operation," doctoral thesis, Cracow University of Technology, 2008.

[12] C.E. Cortés, J. Gibsona, A. Gschwender, M. Munizaga and M. Zúñiga, "Commercial bus speed diagnosis based on GPS-monitored data," Transportation Research Part C 19 (2011), pp. 695–707.

[13] A. Brzeziński and A. Waltz, "Construction of hierarchical traffic models in road network, doctoral thesis, Warsaw University of Technology, 1998.

[14] W.H. Lin and R.L. Bertini, "Modeling Schedule Recovery Processes in Transit Operations for Bus Arrival Time Prediction," Journal of Advanced Transportation, Vol 38, 2004, pp. 347-365.

 ISBN: 978-1-61208-308-7

A System of Pendulums on a Regular Polygon

Alexander P. Buslaev
Moscow Automobile and Road
State Technical University
Moscow, Russia, apal2006@yandex.ru

Alexander G. Tatashev
Moscow University
of Communications and Informatics
Moscow, Russia, a-tatashev@rambler.ru

***Abstract*–We consider a dynamical system, which can be regarded as a transport model. A stochastic and deterministics versions of the model are investigated. The behaviour of the first version of the model is stochastic only at the beginning and over some time becomes a pure deterministic system. The second system comes to a steady state, which depends on the initial state. Considered models can be interpreted as cell automata**.

Keywords-dynamical system; transport models; synergy; cell automata.

I. FORMULATION OF PROBLEM

We consider a mathematical model of a dynamical system. This model can be interpreted as a system of particles, which move in accordance with some rules. The movement of these particles is similar to movement of connected pendulums [1]. The model has also an equivalent interpretation. Namely, the support of movement can be considered as a closed sequence of contours. There are four vertices on each contour, as shown in Fig.1, and a particle, which occupies one of the vertices at each discrete time instant. Each contour has common vertices with two adjacent contours, as shown in Fig. 3. The model can be described as a Markov chain [7]. States of this chain correspond to configurations of particles.

We have obtained mathematical results that concern the behaviour of the system. The cases of small dimensions can be studied by exhaustion. Simulation is used in cases of greater dimensions.

The considered model is similar to a traffic model, which was introduced by K. Nagel and M. Schreckenberg and can be interpreted in terms of cellular automata [2]. Nagel and Schreckenberg investigated the movement on an one-dimensional lattice (straight line or circle).

In Section 1, the considered system is described. In Section 2, a formal description of particles movement rules is given. In Section 3, some propositions are formulated concerning a version of the model with a stochastic rule. In Section 4, some propositions are formulated concerning a version of the model with a deterministic rule.

A contour is considered, which contains four cells NWSE (North, West, South, East). A particle moves on the contour in accordance with rules formulated below. The rings can be joined at points (vertices) NWSE forming a network, as shown in Fig. 1 and Fig. 3.

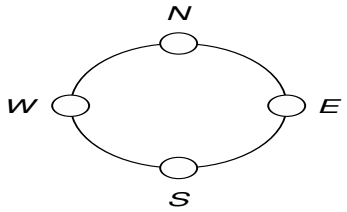


Fig. 1. Basis rings

Let us describe the rules of particles movement on the ring NWSE

(1) *Red state of particle A.* If at present time the cell C ahead of the particle A in the direction of movement is occupied by the particle B of another ring, then the particle A does not move.

(2) *Green state of particle A.* If at present time the cell ahead of the particle A in the direction of movement is vacant and not concurrent, then the particle A comes to C for one step.

(3) *Yellow state of particles A and B.* If at present time the same cell C is the next cell in the direction of movement for two particles A and B (no more particles can be as the network is plane), then, with probability $\alpha = \alpha(A)$, the particle A moves and the particle B does not move, and, with probability $\beta = 1 - \alpha$, the particle B moves and the particle A does not move.

The cell C is called concurrent.

Consider the following systems.

(a) We identify nodes N and S, W and E of the same ring, and we get an elementary "pendulum", Fig 2, $n = 1$.

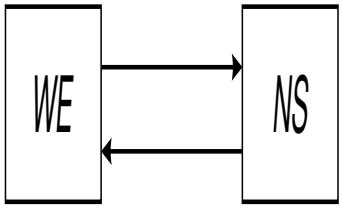


Fig. 2. A ring with joined opposite poles

 ISBN: 978-1-61208-308-7

(b) We consider also a "necklace", i.e., a closed system containing n rings, Fig. 3, $n > 1$.

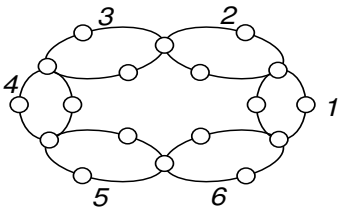


Fig. 3. Necklace

If we identify nodes N and S in each ring of "necklace", then we get a circular pendulum. Round nodes can be occupied successively by particles of the neighboring pendulums, and square nodes can be occupied only by their particle. A pendulum is a side of a regular polygon, as shown in Fig. 4.

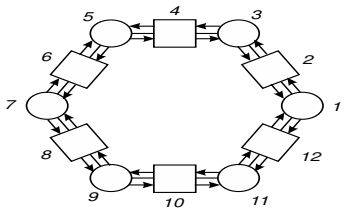


Fig. 4. Circular (hexagonal) pendulum

The main problem considered in the paper is to investigate the system behavior for different possible initial conditions:

(i) state *of synergy,* at which all particles move unimpeded;

(ii) state *of collapse,* at which all particles stop as they cannot move in accordance with the rules;

(iii) *spectrum of system velocities;* each particle, during time T, moves at T_1 steps and does not move at $T - T_1$ steps, or some particles stop and other move permanently.

The considered model is somewhat similar to models investigated in [3- 6], where cellular automata have been used for the local description of the traffic. The difference is that, in present paper, a network model has been introduced.

II. Circular α -n -pendulum

Consider a regular polygon with n vertices, as shown in Fig. 4. The cells are numbered from the vertex E, counter - clockwise $1, 2, 3, 4$, as shown in Fig. 1. Even numbers correspond to main positions and odd numbers correspond to peripheral positions.

States of a particle, on the pendulum k at moment T, are denoted by

$x_k ='' (2k) + 1''$, if the particle occupies the cell $2k$ and moves in direction of $2k + 1$;

$x_k ='' (2k) - 1''$, if the particle occupies also the cell $2k$ but moves in the opposite direction;

$x_k =$ "$(2k + 1) - 1$", if the particle of the k-pendulum occupies at the right peripheral cell and moves back to the cell "$2k$";

$x_k =$" $(2k - 1) + 1$", symmetrically.

Suppose the cell 0 and the cell $2n$ are the same cell. The cell 1 and the cell $2n + 1$ are also the same cell. Suppose $x_1, x_2, x_3, \ldots, x_m$ are the states of particles at present time. Then

(1) $(x_k) \neq (x_{k+1})\ \forall k,\ 1 \leq k \leq n$ as two particles cannot occupy the same cell;

(2) if $(x_k^*) = (x_{k+1}^*)$, then the cell $(2k + 1)$ is concurrent, and the probability of gain are α and $1 - \alpha$; i.e., with probability α realize

$$x_{k+1}^*(T) = x_{k+1}(T + 1),\ x_k(T + 1) = x_k(T),$$

or with $1 - \alpha$ respectively

$$x_k^*(T) = x_k(T + 1),\ x_{k+1}(T) = x_{k+1}(T).$$

(3) if $(x_k) = x_{k+1}^*$, then $x_{k+1}(T + 1) = x_{k+1}(T)$, i.e., the particle $k + 1$ does not move;

(4) if $x_k^*(T) = x_{k+1}^*(T)$, then $x_{k+1}(T+1) = x_{k+1}(T)$, the particle $k + 1$ does not move, i.e., $x_k(T + 1) = x_k(T)$.

III. Some results for α-n-pendulum

The following results have been found for the case $0 < \alpha < 1$.

3.1. *For all initial states, after a time interval with finite expectation, no concurrent cells occur.*

3.2. *For all initial states, the system comes to the state of synergy for a time interval with a finite expectation.*

3.3. *At the state of synergy, the same four states of the system are alternated with period 4.*

3.4. *If $n = 2$, then for all initial states $(T = 0)$ the system comes to the state of synergy no later than at time $T = 2$.*

3.5. *For any T, with non-zero probability, concurrent cells can still appear after time T.*

3.6. *Example one.* Let us fix $E = 1$, $N = 2$, $W = 3$, $S = 4$, as in Fig. 1. Let us consider one direct movement on necklace or equivalent pendulum with $n = 3$. Let (i_1, i_2, i_3) be a state of the system, where $i_j = 1$, if jth particle is at the right position, $i_j = 2$, if jth particle is at the middle position and moves to left; $i_j = 3$, if jth particle is at the left position, $i_j = 4$, if jth particle is at the middle position and moves to right.

Suppose the initial state is $(4, 4, 2)$. The following transitions can be realized

$$(4, 4, 2) \to (1, 4, 3) \to (2, 4, 4) \to (3, 1, 4) \to$$

$$\to (4, 2, 4) \to (4, 3, 1) \to (4, 4, 2).$$

 ISBN: 978-1-61208-308-7

The system has returned to the initial state after 6 steps, as shown in Fig. 5.

3.7. *Example two.* One more example. Consider a dynamical system of type shown in Fig. 3, with $n = 4$ and codirectional movement of particles. State of particle is denoted by letter R, or G, or Y.

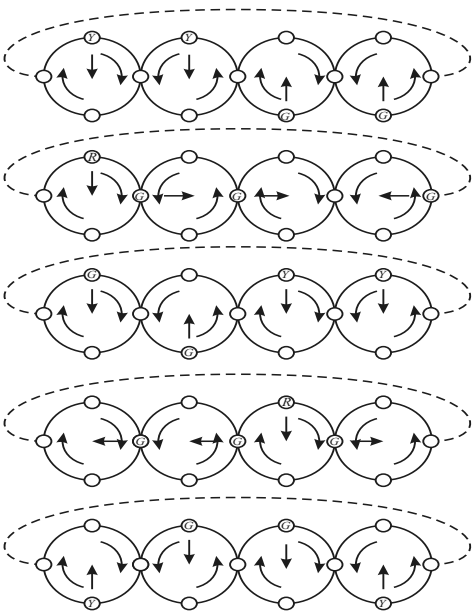


Fig. 5. Nondisappearing yellow color: step by step

Thus, some states with concurrent particles can be repeated with non-zero probability.

3.8. *Synergy effect.* Suppose $n = 3$, and all initial states are equiprobable. With probability 10/13 the system comes to the state of synergy than at time $k = 4$. With probability 3/13 the system comes to the state of synergy after time interval with expectation M

$$M \sim \frac{1}{1-\alpha},\ \alpha \to 1,$$

$$M \to \frac{1}{\alpha},\ \alpha \to 0.$$

In the last case, no finite number k exists such that with probability 1 the system comes to the state of synergy earlier than at time k.

3.9. *Digital synergy.* At the state of synergy, the configuration of particles is defined by position unique one.

IV. Right-priority n-pendulum

Suppose $\alpha = 1$ (analogously, $\alpha = 0$, left - priority). We follow Euler technology [10] of hypotheses burning.

4.1 *Qualitative property.* There are initial states such that the system comes to the state of synergy no later some fixed time, and there are such states that all particles move with same velocities that are less than 1 transition per time unit. For every initial condition, the average velocity of pendulum is greater than 0.5 transition per time unit.

4.2. Let us suppose $n = 2$. For all initial states $(k = 0)$, the system comes to the state of synergy no more than at time $k = 2$.

4.3. Let us suppose that $n = 3$ and all initial states are equiprobable. With probability 23/26 the system comes to the state of synergy no later than at time $k = 4$. With probability 3/26 the same 6 states alternate with period 6. In this case there are 4 transitions of every particle per a period, i.e., the velocity of every particle is equal to 2/3. The expectation of particles velocities is equal to 25/26.

4.4. Let us suppose that $n = 4$ and all initial states are equiprobable. With probability 75/97, the system comes to the synergy for a fixed time, and the same four states alternate with period 4. With probability 22/97, since some fixed time, the states of one of two sets alternate with period 16. In this case, there are 12 transitions of every particle per a period, and the velocity of particles equals 3/4. The expectation of particles velocities is equal to 183/194.

V. Conclusion and future work

We have considered a behaviour of a deterministic system. A stochastic and deterministics versions of the model are investigated.

A "two-dimensional pendulum", which is shown in Fig. 6, will be presented.

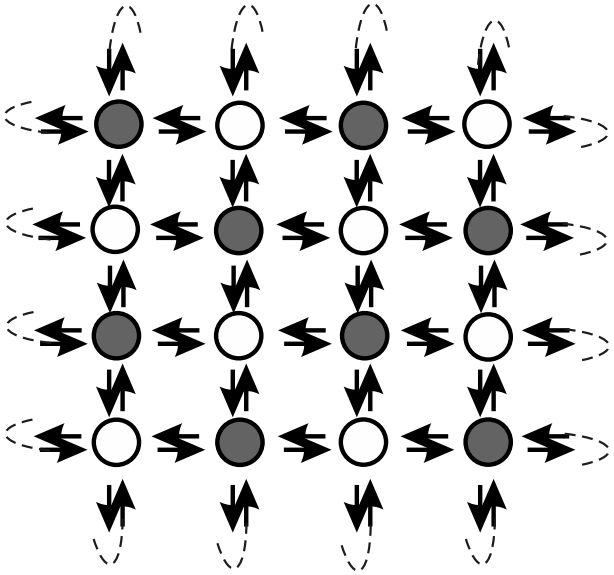


Fig. 6. Two-dimensional pendulum

Vertexes with even sum of row and column indexes, so called "papa-vertexes", contain particles, which move to neighbouring vertexes "mama" according to some plan. One particular case is equivalent to dynamical model of flow on chainmail, as shown in Fig. 7.

 ISBN: 978-1-61208-308-7

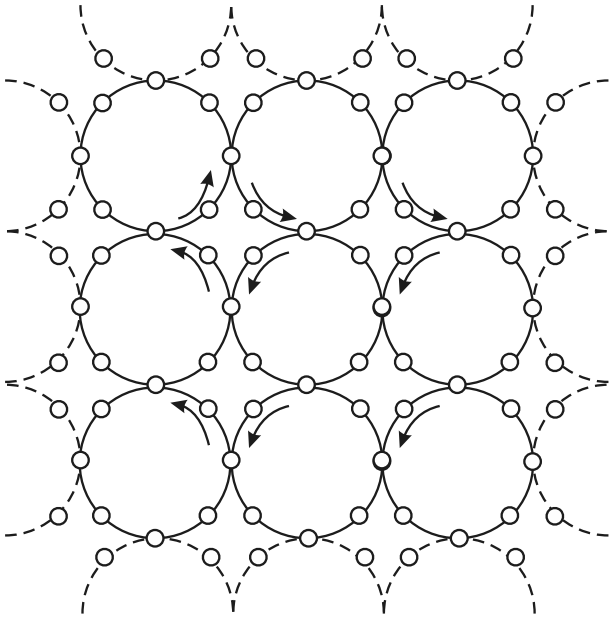


Fig. 7. Flow on chainmail

We also plan to discuss the connection with cellular play of Conway [9].

VI. ACKNOLEDGMENTS

Authors are grateful to postgraduate student A.M. Yaroshenko for creating the computer product and producing computer evaluations [8]. Some computers realizations have been presented. A.M Yaroshenko have developed himself the software and all simulation algorithms.

REFERENCES

[1] L. I. Mandelstam, "Lectures on waves", 1930 - 1932 Polnoe sobranie trudov vol. 4, Izd. AN SSSR, Moscow, 1955.

[2] K. Nagel and M. Schreckenberg, "A cellular automaton model for freeway traffic", J. Phys. I France 2, 1992, pp. 2221-–2229.

[3] V. V. Kozlov, A. P. Buslaev, , and A. G. Tatashev, A. G., "Monotonic random walks and clusters flows on networks. Models and applications", Saarbruecken: Lambert Academician Publishing, 2013)

[4] A.P. Buslaev A.P. and Tatashev, A.G. "Particles flow on the regular polygon", Journal of Concrete and Applicable Mathematics (JCAAM), 2011, vol. 9, no. 4, pp. 290–303.

[5] Buslaev, A.P., Tatashev, A.G., and Yashina M.V., "Cluster flow models and properties of appropriate dynamic systems", Journal of Applied and Functional Analysis (JAFA), 2013, vol. 8, no. 1, pp. 54–76.

[6] Kozlov,V.V., Buslaev A.P., and Tatashev, A.G. "On synergy of totally connected flows on chainmails", Proceedings of the 13th International Conference on Computational and Mathematical Methods in Science and Engeneering (CMSSE-2013), Almeria, Spain, June, 24–28, 2013, v.3 , pp. 861-874

[7] Feller, A. "An introduction to probability theory and its applications", New York: John Wiley, 1968.

[8] Yaroschenko, A.M. "Simulation models of monotone random walks on graphs", Proceedings of the 13th International Conference on Computational and Mathematical Methods in Science and Engeneering (CMSSE-2013), Almeria, Spain, June, 24–28, 2013, v.4 , pp. 1438-1449

[9] Berlekamp, E., Conway, J., and Guy, R. "Winning ways for your mathematical plays", London: Academic Press, 1982.

[10] Borodin, A.I, Bugay, A.S. "Biographical Dictionary of Mathematicians.", Kiev, Radyanska Shkola, 1979 (in Russian).

 ISBN: 978-1-61208-308-7

Concept for a Task–Specific Reconfigurable Driving Simulator

Bassem Hassan
Heinz Nixdorf Institute
University of Paderborn
33102 Paderborn, Germany
Bassem.Hassan@hni.upb.de

Jürgen Gausemeier
Heinz Nixdorf Institute
University of Paderborn
33102 Paderborn, Germany
Juergsen.Gausemeier@hni.uni-paderborn.de

***Abstract* - Driving simulators support the development process of new vehicle systems, such as Advanced Driver Assistance Systems (ADAS). Driving simulators are varying in their structural complexity, fidelity and cost. Furthermore, they consist of numerous simulation models that cooperate at runtime. These partial models represent dedicated aspects of the vehicle under test, other traffic participants as well as the environment. Since the development of a driving simulator is costly and complex task, testing and training of ADAS often requires more than one driving simulator. There is a need for a reconfigurable driving simulator that allows the operator to reconfigure the simulator and its structure or exchange the simulation models in a simple way without in depth know how of the system structure or interface topology. This paper describes a concept for a task-specific reconfigurable driving simulator for testing ADAS. A systematic for the development process is presented, the key software and hardware components of the driving simulator are identified, and a configuration mechanism and its software are briefly presented.**

Keywords - Advanced Driver Assistance Aystems (ADAS); reconfigurable driving simulator; confiuration mechanis; solution elements

I. INTRODUCTION

The influence of modern Advanced Driver Assistance Systems (ADAS) on the vehicle gets more and more complex making it increasingly difficult to understand and hence analyze the interplay between ADAS, vehicle, vehicle environment and driver. Driving simulators have played a vivid part in developing new automobiles and their subsystems, providing reproducible testing conditions and a safe testing environment, as well as a means to reduce development time and cost.

Nevertheless, driving simulators are typically designed and built for a special purpose in order to support a specific analysis task in a predefined environment. Adapting such systems to support new functions or applications is very complex, time consuming, and often not feasible. Thus, with the increasing role of ADAS, there is a need for highly adaptable and reconfigurable systems that can be conveniently tailored to new specific functions.

Such a reconfigurable driving simulator closely resembles a building block concept. The various simulation models, software and hardware components that constitute the driving simulator need to be widely combinable. Moreover, there are also different levels of details simulation models, ranging from simple low-fidelity models to their respective complex high-end models, which provide a detailed simulation. Also the hardware components range from simple to complex components, which constitute different simulator setups that are capable of simulating specific aspects of the ADAS under test [1].

A reconfigurable driving simulator requires compatible software and hardware interfaces and a reliable checking mechanism to achieve consistent configurations of the system [1].

The second section of this paper, will describe the state of the art and the related work. The third section will describe the reconfigurable driving simulator concept. The fourth section will be a conclusion and the future work.

This work is part of the project TRAFFIS (German acronym for Test and Training Environment for Advanced Driver Assistance Systems).

II. RELATED WORK

Modern driving simulators show a broad range of applications, e.g., driver and safety training, vehicle evaluation, road design, and vehicle dynamics simulation. They can be used for research purposes to study the behavior of the driver, develop and evaluate new vehicle subsystems such as ADAS, driver training and many more. Furthermore, driving simulators range from expensive from low-cost desktop systems, which are even used in small companies, to high-end systems, which can realistically affect an entire vehicle mounted on a motion platform. Due to the big variation of the driving simulators complexity, fidelity, cost and the purpose of use; the conception, specification and selection process of a driving simulator is a hard process [2-4]. Typically the driving simulator user is not an expert in driving simulator techniques and can't decide easily which driving simulator match with his requirements. Methods for selecting a driving simulator have been developed, for example, Negele [5] proposed a method that allows the developer of new vehicle technologies to formulate and customize a driving simulator concept to match his requirements, in form of advice which subsystem has to be used in which application scenario.

Here are also three examples of previous attempts towards building a reconfigurable driving simulator:

 ISBN: 978-1-61208-308-7

A. *UCF (University of Central Florida) Driving Simulator*

The UCF driving simulator housed in the Center for Advanced Transportation Systems Simulation; is a driving simulator with a high driving fidelity and immense virtual environments. The UCF driving simulator provides a research platform for multi-disciplinary investigations in the traffic engineering area [6].

The UCF simulator was designed, that the vehicle simulation model and the vehicle mock-up are interchangeable [7].

B. *University of Valencia Training Simulator*

The Valencia training simulator is a cranes driving simulator for training of new workers and operators of port areas that are responsible for loading and unloading ships [8].

The Valencia driving simulator was designed, that the crane simulation model and the projection system are interchangeable [8].

C. *BVG Trams Driving Simulator*

The "Berliner Verkehrsbetriebe" (BVG) is the widest tram network in Germany [9], the BVG simulator providing a training platform for the trams driver.

The BVG simulator was designed so that the tram mock-up is interchangeable.

There were three examples of simulators, which are giving some configurability or flexibility by exchanging one or more components of driving simulators, these examples shown that till date there is no method or approach, which allow the user to reconfigure whole the system without in depth know how of the system structure and components.

III. RECONFIGURABLE DRIVING SIMULATOR CONCEPT

Towards a reconfigurable driving simulator, there is a need for a development systematic. It describes how to build up a reconfigurable simulation system. The main concept of the development systematic is described in form of phases/milestones diagram as shown in Fig. 1. It is divided into the following five phases/milestones:

A. System specification
In this phase the driving simulator will be specified by using specification technique. The specification technique itself will be shortly described and the important specification results e.g. the application scenarios will be presented.

B. Solution elements deployment
In this phase the existing solution elements e.g. different variants of the vehicle mock-up will be consolidate into the reconfigurable system. The concept of the solution elements deployment will be presented.

C. Configuration mechanism development
In this phase a configuration mechanism will be developed. This ensures the consistency and compatibility of the selected solution elements from the above step, to build a valid configuration setup of the driving simulator.

D. Configuration tool development
In this phase a user friendly interface will be designed and developed, to allow the user easily to select the suitable solution elements and to generate a driving simulator configuration.

E. Simulation initialization
In this phase the interface between the generated configuration and the simulation software will be developed.

On the following part, each phase/milestone, as well as the results of each phase/milestone, will be described.

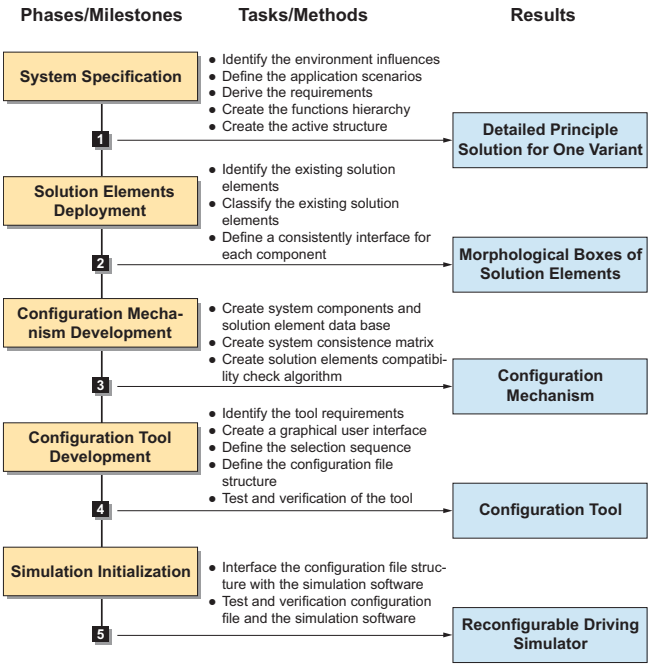


Figure 1. Reconfigurable driving simulator development systematic

A. *System Specification*

Driving simulators are complex multidisciplinary systems, which consist of mechanical components; such as the motion platform, electronics components e.g. microcontrollers, control components e.g. motion cueing algorithms, and Information technology components; such as various software and simulation models. Accordingly, a driving simulator with a motion platform is considered as a typical mechatronic system.

In order to describe a mechatronic system in early design phases, which is called "Principle Solution", the Conceptual Design Specification Technique for the Engineering of Complex Systems "CONSENS" is used [10], which was developed at the University of Paderborn.

As shown in Fig. 2, the developed specification technique divides the description of a principle solution into

 ISBN: 978-1-61208-308-7

different partial models according to specific aspects as following: requirements, environment, application scenarios, functions, active structure, shape and behavior [10].

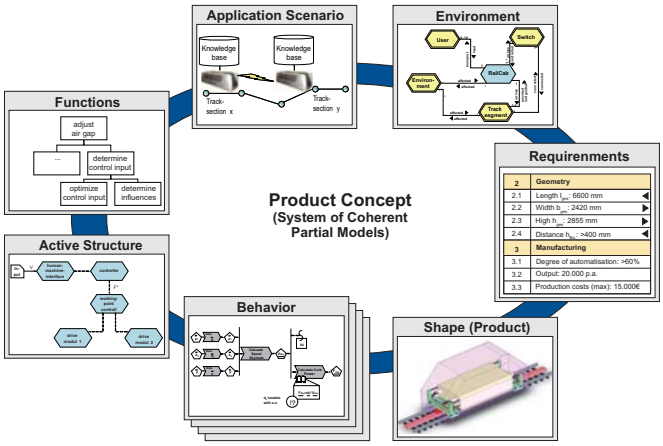


Figure 2. Specification of coherent partial models according to CONSENS [10]

We used CONSENS to develop the driving simulator principle solutions; this modeling process was done during many workshops with the driving simulator developers and users, who are involved in the TRAFFIS project. The important modeling results will described in the next part.

Environment: Investigation of the driving simulator environment shows that the external influences come from: driver, driving simulator operator, driving instructor/test manger, energy source and the ground.

Application scenarios: As described in the introduction; the TRAFFIS project focuses on testing and training for ADAS. The ADAS example used here is a Headlamp Control Module (HCM). HCM is a driver assistance system developed by Varroc [11]. It controls the headlamps to ensure an optimum road illumination by using high beam as often as possible. The project partners defined 7 application scenarios, which were classified and summarized into 3 different simulator setups as following:

- TRAFFIS Full Version: This application scenario has the most complex structure with a full scale motion platform. Its target is to test an ADAS control unit and the camera as Hardware in the Loop (HiL) simultaneously together with a Driver in the Loop (DiL).
- TRAFFIS Portable Version: This application scenario is a stripped version from the full version; its target is to train truck drivers on the new ADAS.The training should be by the logistics centers onsite, therefore a portable system with simple motion platform is needed.
- TRAFFIS Light Version: The aim of the application scenario is to test ADAS as Software in the Loop (SiL). Here, only a computer based system is utilized without a motion platform. This version is useful for the ADAS developers in early phases to test his control algorithms daily in a simple way.

Active structure: Since the full version is the most complete simulator configuration, it is modeled in this partial model in order to identify the driving simulator main components and the relationship between these components, with help of the defined application scenario. The result of the active structure modeling will be presented next and it is shown also in a simplified way in Fig. 3; they are categorized into the well-known model-simulation-analysis process.

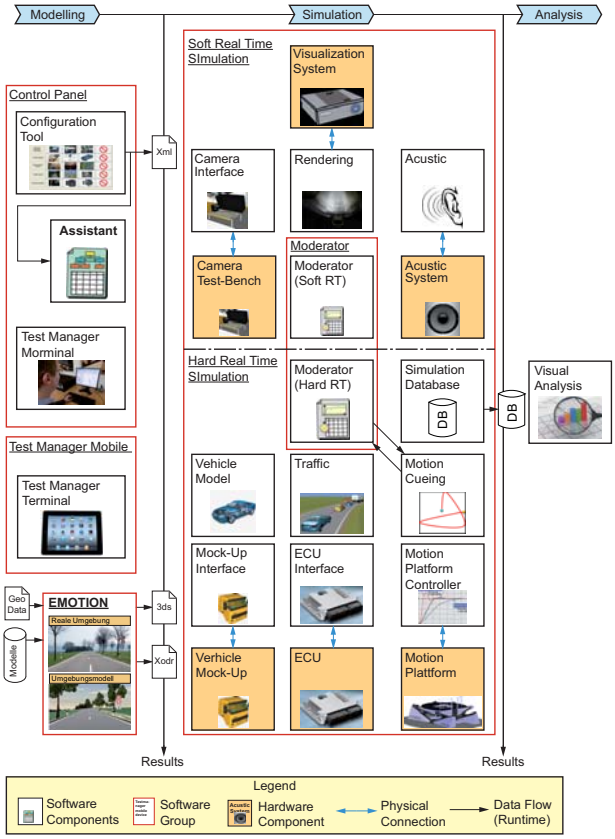


Figure 3. Identified driving simulator system components

The main result from the first phase is the identification of the driving simulator main components within in the principle solution; the important identified components are described as following:

Emotion (software): This is a software tool that allows the automatic generation of virtual roadways based on geographic information systems [12]. The software generates logic and graphic representation of the environment.

Configuration tool (software): Target of this tool is to create a driving simulator setup depending on the user needs by selecting the desired software and hardware components to generate a task specific driving simulator setup. This software concept will be described in detail in section C and D.

 ISBN: 978-1-61208-308-7

Assistant (software): This software operates in a preprocessing step, reads the configuration file, allocates selected software components, loading them, then distributes them over the available resources (computers, real-time hardware, etc.).

Operator (software): The operator software is a user-friendly graphical user interface, which enables the driving simulator operator to operate (load, start, stop, etc.) the whole system by using a standalone software.

Test manager mobile terminal (software): This software component runs on a mobile device, it's allow the test manager or the driving instructor to trigger some predefined events in run time.

Moderator (software): The moderator plays a very important rule during the simulation run-time. It reads the configuration file, which contains the communication and interface topologies between the different software components and establishes the communication between these components regarding their real-time specifications.

Vehicle mock-up (hardware): This is the real vehicle mock-up (driver's cabin), which represents the MMI (man-machine interface) between the driver and the driving simulator.

Vehicle mock-up (software): This software interface redirects the input from the dashboard in the driver cabin to the simulation software and vice verse the output from the simulation to the driver cabin.

Vehicle model (software): It is a software package developed to simulate the actual nonlinear physical characteristics of a typical vehicle in response to the operator inputs in real-time using a multi-body dynamics system.

Motion cueing algorithm (software): This describes the presentation of haptic information (cues) with the aim to resemble real movements in virtual environments [13].

Motion platform (hardware): Is an active mechanism, which allows haptic feelings of being in moving.

Motion platform controller (software): In order to synthesize a controller for actuators of motion system, different actuator controllers can be used to guarantee that the motion system of the simulator exactly follows a desired trajectory defined by the motion cueing algorithm [14].

Rendering (software): The rendering software visualizes the virtual driving scenario onto the projection system. It typically renders the driver view showing all detail of the current driving situation, as well as other traffic participants, the road, and the complete environmental scenery around the simulated vehicle.

Visualization system (hardware): The rendered scenery needs to be presented to the driver in the vehicle. Thus the rendering software feeds different types of visualizations devices such as screens or projectors.

Simulation results (database): During a driving session, some selected data are logged and stored in a database for later retrieval and analysis.

Visual Analysis (software): The logged simulation results are later used in the post processing phase for a visual analysis of the driving session.

The following components will be only listed but not described:

Traffic model (software), Control unit (hardware), Control unit (software), Camera test-bench (hardware) [15], Camera test-bench (software) [15], Acoustic (software) and Acoustic system (hardware).

B. Solution Elements Deployment

The result of the first phase is the identification of the main driving simulator components. Each component defines a function of hardware or software subsystem and its standard inputs outputs interfaces. Under each component is a group of possible solutions, which are called solution elements.

For example, the vehicle model is a software component defines the physical model of the vehicle under test, and the group of the solution elements could be (personal car model from dSPACE Company, personal car model from Paderborn University, truck model from dSPACE Company, etc.).

In order to consolidate all existing solution elements into the reconfigurable driving simulator, the existing solution elements have been identified and classified under its corresponding components; moreover a general interfacing concept had been developed.

To interface the entire solution elements together without in depth know how of each solution element, each solution element is considered as a black box. That means that only the inputs and outputs interfaces have to be taken in account. To keep the configuration process flexible and extendable any solution element could be added as soon as his inputs and outputs interfaces are defined. The only required task to integrate any solution element is to map its inputs and outputs to the predefined inputs and outputs of the parent component, which is called here signal multiplexing.

Fig. 4 shows an example of this concept: a vehicle model has to be integrated as a solution element, the model will be considered as a black box, but all the signals has to be mapped to the parent component signal description. The output signal which called "Otutput_ID563[m/s]" contains the vehicle under test velocity in m/s, but this signal unique name and unit predefined by the parent component is "Chasis_Velocity" in km/h.

In order to integrate this vehicle model; the user has to connect all the input and output signals with different names and units to the unique names and units of the parent component.

This concept allows the user to integrate new solution elements to the system with minimum effort, since this integration step will be done only once for each new solution element.

 ISBN: 978-1-61208-308-7

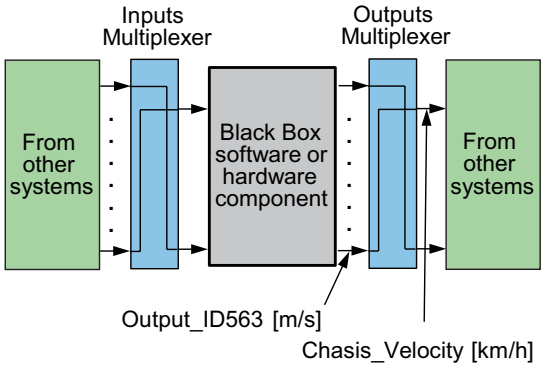


Figure 4. Integration of a Black Box model into Driving Simulator System

Since all solution elements are integrated within the configuration software, the result is a driving simulator configuration matrix shown in Fig. 5. The first column of the matrix lists all 23 components as identified in the first phase, while the rows contain different solution elements for each component. Fig. 5 shows the configuration matrix in form of morphological boxes.

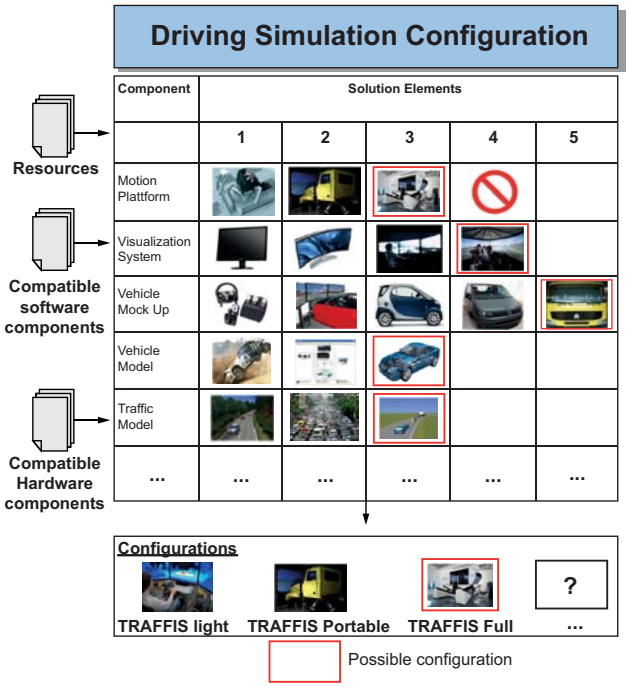


Figure 5. Driving simulator morphologische boxes

C. Configuration Mechanism Development

The result of the second phase is the configuration matrix in morphological boxes form. The morphological boxes allows the user to select one solution element under each row "component" and the combination of these selected solutions elements described a possible configuration. To ensure the consistency and compatibility of the selected solution elements, a configuration mechanism has to be developed to ensure the consistency and compatibility.

The configuration mechanism is ensuring the following function: preservation of the selected solution element consistency, checks the solution elements compatibility, add or modify components and solution elements and generating the configuration file, which contains the selected components, interface topology, selected resources, and signals for analysis. This mechanism concept and a prototypical implementation are done during master thesis supervised by the authors.

D. Configuration Tool Development

The result of the third phase is the configuration mechanism, which is a complex sequential process. In order to make a user friendly interface, in this phase, a concept and implementation of easy to use software has to be developed. The software guides the user during the selection process in a sequential order till the user finishing the configuration. The advantage of such software is to execute the configuration mechanism in the background without any annoying of the user. Fig. 6 shows a screen shot of the developed software.

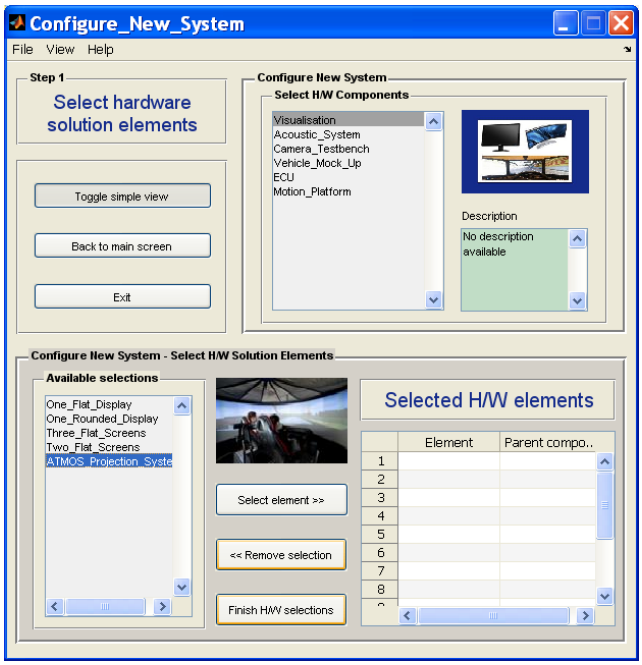


Figure 6. Configuration software

E. Simulation Initialization

The result of the fourth phase is the configuration file. The configuration file contains the selected components, interface topology, selected resources, and signals for analysis. In order to use the generated configuration file, there is a need to interface this configuration file with the simulation software. The interfacing between the configuration file and the simulation will done by two software components “the assistant” and “the moderator”, the interfacing will done through following steps:

Software distribution over resources: The assistant software reads the driving simulator configuration file; it retrieves the models, software components from the file storage system and distributes them over the available resources (computers).

Software initialization: The moderator software initializes the communication between all system components as described in the interface topology in the configuration file and prepares the system for start-up.

Simulation run-time: as soon as the user starting the simulation, the moderator software ensures the communication between all the system components and logs the selected signals for analysis in the configuration file during run time in the simulation database.

Post-processing: After the simulation session is finished, the moderator software logged signals from the simulation run-time for analysis.

Regarding the interface topology (moderator task on run-time), it is a challenge to connect all 23 components together. This work is often done manually by connecting input and output signals of the components directly. Moreover, manual connection has to be repeated or modified each time the driving simulator structure is changed to fit user specifications. Therefore, there is a real need for an automated method in order to moderate the interface between the selected solution elements, to allow the user to change the system structure or to replace one or more solution element without knowing details or the interface topology.

The concept of the communication topology is that all components will be connected by the moderator software via an input signal bus and an output signal bus instead of connecting the system components directly. Fig. 2 shows an example of the communication between the moderator software and the motion cueing component (two black arrows). The signal bus contains a detailed description of each signal it contains: unique signal name, signal unit, frequency, resolution, communication protocol, physical port, if this signal is mandatory or optional, and a description of the signal.

The moderator software ensures the communication between all the system components, as shown in Fig. 7. Moreover, the moderator software checks the selected signals for analysis in the configuration file, and logs them during run time in the simulation database.

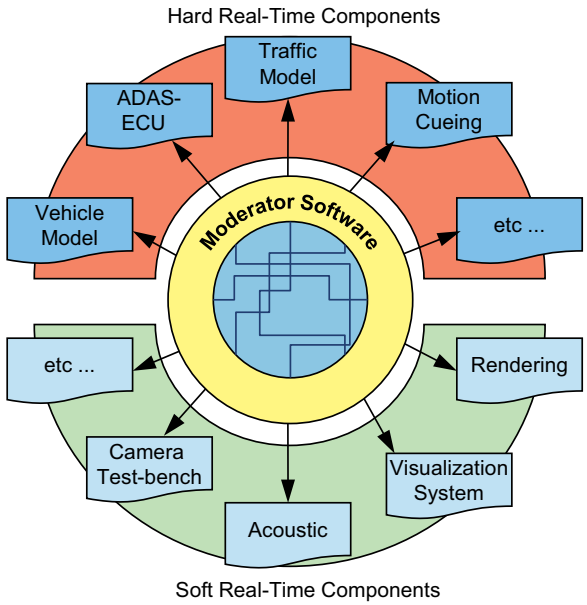


Figure 7. Moderator Software Task at run-time

IV. Conclusion and Future Work

This paper described a concept for a task-specific reconfigurable driving simulator for testing and training of ADAS. The concept demonstrates how individual subcomponents and their respective simulation models of varying fidelity and level of detail are configured into valid and working simulator setup based on the requirements of previously defined in the application scenarios. We applied the concept with the context of the TRAFFIS project and achieved promising results with our first reconfigurable simulator two prototypes (*TRAFFIS Full Version and TRAFFIS Light Version*). These prototypes vary distinctively from each other depending on the different application focuses for test and training of ADAS in their respective testing environments.

Within the TRAFFIS project context; we plan to further prove this concept by developing a third simulator prototype and elaborate more complex application scenarios for testing and training of further functions of ADAS in more complex environments.

Acknowledgment

This work, as part of the project TRAFFIS (German acronym for “Test and Training Environment for Advanced Driver Assistance Systems”), which is funded by European Union “ERDF: European Regional Development Fund” and the Ministry of Economy, Energy, Industry, Trade and Craft of North Rhine Westphalia – Germany, within the “Ziel2” program.

We thank our project partner dSPACE for providing detailed vehicle and traffic models, as well as specific HiL-simulation hardware. We thank our project partner Varroc for providing a head light control module for a adaptive bending lights.

REFERENCES

[1] B. Hassan, J. Berssenbrügge, I. Al Qaisi, and J. Stöcklein, "Reconfigurable Driving Simulator for Testing and Training of Advanced Driver Assistance Systems," Proc. International Symp. on Assembly and Manufacturing (ISAM 2013), July 30th – Aug 2th, 2013, Xian, China.

[2] K. Koscher, A. Czeskis, F. Roesner, S. Patel, S. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental Security Analysis of a Modern Automobile," Proc. IEEE Symp. on Security and Privacy, May 16-19 2010, Berleley/Oakland, California, USA.

[3] G. Weinberg and B. Harsham, "Developing a Low-Cost Driving Simulator for the Evaluation of In-Vehicle Technologies," Proc. of the First International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI 2009) Sep 21-22 2009, Essen, Germany.

[4] J. Berssenbrügge, "Virtual Night Drive – A Method for Displaying the Complex Light Distribution Characteristics of Modern Headlight Systems Within a Simulated Night Drive," Ph.D. thesis, Faculty of Mechanical Engineering, 2005, University of Paderborn, Germany.

[5] J. Negele, „Anwendungsgerechte Konzipierung von Fahrsimulatoren für die Fahrzeugentwicklung," Ph.D. thesis, Faculty of Mechanical Engineering, 2007, Technische Universität München, Germany.

[6] M. Abdel-Aty, X. Yan, and E. Radwan, " Using the UCF Driving Simulator as a Test Bed for High Risk Locations," Technical Report: Florida department of transportation, 2007, Florida, USA.

[7] D. Gue, H. Klee, and E. Radwan, "Comparison of Lateral Control in a Reconfigurable Driving Simulator," Proc. DSC North America, 2003, Dearborn, Michigan, USA.

[8] M. Lozano, M. Fernandez, and R. Martinez,"A Reconfigurable Projection System For Several Gantry Crane Simulators," Technical Report, 1999, University of Valencia, Spain.

[9] http://www.bvg.de/index.php/en/17106/name/Tram.html [retrieved: October, 2013]

[10] J. Gausemeier, T. Gaukstern, and C. Tschirner, "Systems Engineering Management Based on a Discipline-Spanning System Model," Proc. Conference on Systems Engineering Research (CSER'13), Eds.: C.J.J. Paredis, C. Bishop, D. Bodner, Georgia Institute of Technology, March 19-22, 2013, Atlanta, USA.

[11] C. Schmidt, "How to Make an AFS System Predictive: ADASIS Interface Implementation," Proc. 7th International Symposium on Automotive Lighting, September 25-26, 2007, Darmstadt, Germany.

[12] S. Kreft, J. Gausemeier, M. Grafe, and B. Hassan, "Automated Generation of Virtual Roadways based on Geographic Information Systems," Proc. International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (ASME 2011), August 29 - 31 2011, Washington DC, USA.

[13] J. J. Slob, "State-of-the-Art Driving Simulators," a Literature Survey, 2008, Eindhoven University of Technology, Eindhoven, Netherlands.

[14] I. Al Qaisi and A. Trächtler, "Constrained Linear Quadratic Optimal Controller for Motion Control of ATMOS Driving simulator," Proc. Driving Simulation Conference, 2012, Paris, France.

[15] Y. Tan and B. Hassan, "A Concept of Camera test-bench for testing Camera Based Advanced Driver Assistance Systems," Proc. International Design Engineering Technical Conferences & Computers and Information in Engineering Conference (ASME 2013), August 4 - 7 2013, Portland, USA.

 ISBN: 978-1-61208-308-7

Simulation and Validation of a Heuristic Scheduling Algorithm for Multicore Systems

James Docherty, Alex Bystrov, Alex Yakovlev
School of EEE
Newcastle University
Newcastle-upon-Tyne, UK
james.docherty/a.bystrov/alex.yakovlev@ncl.ac.uk

***Abstract*—The number of embedded systems used worldwide is increasing rapidly. With each generation of equipment, consumers are expecting more computational power and functionality, meaning current designs can be considered unsuitable. As transistor feature size reaches its atomic limit, manufacturers have moved from single to multi-core environments to bridge the performance gap and continue to meet Moores Law. However, this means job scheduling has become exponentially more complex and is reaching a point where standard algorithms are failing to cope. This paper summarizes the initial work performed creating a heuristic based algorithm that is aware of both requests and available resources and therefore is capable of managing the uncertainty brought about by these factors. The work uses Monte Carlo Simulation combined with multi-vary analysis to identify primary contributors and their contribution to scheduling.**

Keywords-Energy Harvesting; Heuristic Algorithms; Monte Carlo Simulations;

I. INTRODUCTION

With the increase in pervasive computing across the world, ever more embedded systems work primarily on stored energy to perform computationally intensive tasks. This means management of both jobs and power becomes more important in every design iteration. A major technique for managing dynamic power dissipation (power consumed by active switching) that is becoming widely used is Dynamic Voltage Frequency Scaling (DVFS). This reduces power consumption at the penalty of increasing execution time. In a system where task execution is short, but with regular activation times, this technique could reduce power consumption by up to 30% [1]. However, many multi-core systems now operate close to their critical voltage, meaning reduction of core voltage to reduce energy consumption is not possible [2]. Therefore more intelligent management of the jobs presented must take place to maximize functionality while conserving reliability.

As performance continues to be a key metric, microprocessor designers aim to meet the goals set by Gordon Moore in 1965 that the number of transistors on an integrated circuit would double every 18 to 24 months [3]. This goal, now known as "Moores Law" is now seen by designers as a target to be met and has been achieved by a number of processors [4]. While this was originally done by decreasing feature size, as transistors head below 10nm, we are approaching the atomic limit; where leakage current is too great to be acceptable for normal use. Therefore, the design of systems rather than devices has changed, with multi core environments allowing Moores predictions to remain true. However, as the number of cores increase, the complexity of managing them grows exponentially. Standard scheduling, working on such methods as First Come First Served or Priority Queuing can cope while the core number is small, but will eventually fail, especially once threads with precedence and relations are present [5]. Eventually these problems will become NP-Hard and feasible schedules will fail to be constructed in a timely manner.

Heuristic Algorithms may offer a solution to this problem, by having simple rules based on factors such as job arrival rate and available energy and using designer intuition,. One such method of heuristic planning is Game Theory which, since its development by Von Neumann in the 1940s [6] and subsequent work by John Forbes Nash [7], has found use in many fields from computer science to evolutionary biology, as well as power management [8] [9].

The models developed tend to take on simple rules, which over time lead to a stabilized outcome. Ideally this outcome will maximize for all players, making the game pareto optimal. This way, if a sudden change occurs, the game will become unstable and eventually settle over a set of repetitions. These games can be modelled as either cooperative or non-cooperative, with much research concentrating on non-cooperative models where every player is competing to maximize their payout. However, some work [10] suggests cooperative games could also lead to greater power savings against the penalty of a power management kernel. Within a system, this game would have many players and input variables, therefore a method must be developed that can reduce these into an optimal model and manage scheduling based on key parameters only.

This paper presents the simulation testing of a Heuristic Algorithm developed in MATLAB to compensate for these issues. The Hypothesis is that a simple algorithm aware of parameters such as overall system load should outperform standard queuing paradigms and increase operational life. Section II details the processor design, queuing methods and simulations conducted, along with techniques used to refine the Heuristic Algorithm. Section III analyses the outcomes from the MATLAB simulation and optimisation runs, discussing the findings. Section IV summarises findings and Section V proposes further investigation for these outcomes.

II. METHODOLOGY

A. Job Queuing

To develop a heuristic based algorithm, a simplified system was modelled in MATLAB. This took the form of Figure 1, with jobs arriving and being sent to cores by a scheduler.

 ISBN: 978-1-61208-308-7

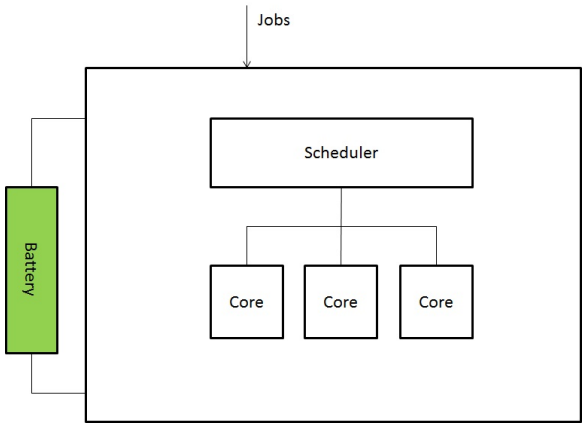


Fig. 1. Simplified Microprocessor Design for Simulations

The cores would manage these jobs and inform the scheduler when they are available to accept new tasks. Three models were chosen for testing. These were:

- First Come First Served (FCFS)
- Priority Queuing
- Heuristic Algorithm under investigation.

First Come First Served is the simplest, but often most effective method of queue management [11]. One queue exists, which all jobs enter and the job at the head of the queue is serviced. This is feasible to implement, while also giving good performance for low workloads. However, as the workload increases, queues can build to extreme lengths and lead to issues.

Priority Queuing adds a second queue for high priority jobs. This is similar to the system seen at airports where check-in desks will serve the main queue until someone joins the fast-track queue. This reduces issues with FCFS when jobs have varying priorities, but can lead to large volumes building up in the main queue if priority jobs continue to block the processor either through high arrival rates or processor utilization.

Both these and other scheduling algorithms fail to dynamically manage cores to maximize system reliability. Many also lack the ability to manage precedence and are unaware of energy as an input. The proposed solution in this paper provides these important items, while also managing job direction through simple heuristics, rather than a complex management program.

B. Literature Review

Heuristics have proved a popular method of managing jobs in a variety of areas, especially where a level of non-determinism is present, as this can lead to exhaustive search methods becoming excessively long in their computation time. Methods can use explicit rules to find optimal strategies [12] or more abstract techniques such as Nash Equilibrium discussed previously [13]. Heuristics allow for schedules that are clearly infeasible to be ignored through initial grouping, which reduces computation time and can improve performance [14] [15]. This use of heuristics means some scheduling that would originally have been performed offline can now take place in real-time, improving the reliability of operation [16]. As complexity of systems increases, the use of rule-based schedulers will become more commonplace. While these cannot always give the best response, the outcome will often be suitable in the time taken to calculate it that no detriment to the quality of service will perceptibly take place [17].

C. System Level Simulation

An algorithm, which can be seen in Algorithm 1, was created to allow System Level Simulations to take place. This let a series of jobs be run through a Monte Carlo Simulation. Jobs are created based on a rate (λ) and processed based on another rate (μ). One to three cores can also be implemented, allowing the difference in queue times, queue length and processor utilization to be observed. For this jobs are assumed to be arriving from activation (t_0) at a rate of $\lambda e^{-\lambda t}$ and processed at a rate of $\mu e^{-\mu t}$. The simulation is run over 72000 cycles with 10 repeats for each case and λ/μ values of 10, 30 and 50. One, two and three cores are active in versions of the simulation and all simulations are repeated ten times to give consistency of results.

Once job arrival and service rates are determined in lines 4-5, the system enters the 72000 operational cycles. The algorithm considers available energy prior to scheduling and adjusts the level to activate the Heuristic section through variable $MaxQueueLength$. Cores are only activated to service jobs if the current queue length is greater than this variable, preventing the excessive consumption of energy but increasing total operation time for jobs. If all cores are active and a job is low priority, it will be placed into the queue at the tail, replicating FCFS. However, priority jobs will be placed into the priority queue, which will be serviced by the next available core. All data was outputted to a CSV file on completion of 72000 cycles or if system energy reached zero during simulation when this was considered.

Data from the simulations was collected and analysed using Minitab, a statistical program used for data analysis, to look for statistical differences between scheduler types and key parameters to be used in any improvement exercises.

D. Addition of Energy

Once these experiments were complete, giving a baseline of algorithm effectiveness, the extra uncertainty of energy was added to the simulations. This was done by placing a battery into the simulation with a percentage of charge. The FIFO and Priority Queues were unaware of this and therefore continued executing jobs at the same rate until failure. However, as the state of charge decreased, the aggression of the heuristic algorithm in activating cores decreased through $MaxQueueLength$, meaning the execution time for jobs increased. While this would seem to decrease the quality of service, the goal is to maximise lifetime of the system. Since fewer cores are activated, both the static and dynamic energy consumptions are decreased — thereby increasing the operational life of the system. Priority queuing is still active

 ISBN: 978-1-61208-308-7

Algorithm 1 Simulation for Heuristic Scheduler

```
Set λ and μ
Set Flags and Clock to zero, Energy to 100
for ArrivalTime and ServiceTime = 1 to 72000 do
    ArrivalTime(n) = λe^{-λt}
    ServiceTime(n) = μe^{-μt}
end for
while Clock < 72000 do
    if Energy < 50 then
        MaxQueueLength = 2
    else
        MaxQueueLength = 4
    end if
    for AllActiveCores do
        ServiceTime − −
    end for
    if QueueLength > 0 then
        if CoreisEmpty & CoreIsActive then
            PlaceJobonCore
        else if CoreisEmpty & CoreisInactive & QueueLength > MaxQueueLength then
            ActivateCore, PlaceJob
        end if
        if Clock = ArrivalofNewJob then
            if CoreisEmpty & CoreisActive then
                PlaceJobonCore
            end if
        else if CoreisEmpty & CoreisInactive & QueueLength > MaxQueueLength then
            ActivateCoreandPlaceJob
        else if AllCoresAreBusy & JobIsNonpriority then
            PlaceJobinQueue
            QueueLength + +
        else if AllCoresAreBusy & JobIsPriority then
            PlaceJobinPriorityQueue
        end if
    end if
    for EachActiveCore(1 − x) do
        Core(x)EnergyConsumed = U(0.002, 0.005)
    end for
    Energy = Energy − AllCore(x) EnergyConsumed − N(0.01, 0.005)
    if Energy <= 0 then
        Break
    end if
    UpdateWaitTime, IdleTime, QueueLength
    Clock + +
end while
print Value of Energy, Cores, λ, μ, Average Wait Time, Max Wait Time, Average Queue Length, Max Queue Length, Clock, Cores 1-3 Idle Percentage
```

and priority jobs will be fast tracked to the core, with jobs currently occupying it halted. Once all priority tasks are cleared from the system, the non-priority jobs may resume executing. In extreme cases, the scheduler may activate a core to push through more priority jobs, thus maintaining quality of service for a small energy penalty. Though this will shorten the operational life of the entire system, missing priority jobs could be hazardous, especially in real-time or safety critical systems. Therefore this reduction in system lifetime is justified by keeping key systems active.

E. Design of Experiments

Following this preliminary work, a Design of Experiments (DoE) was conducted to determine each variables overall contribution and whether any interaction between variables took place. Within a DoE, key parameters and their values are run to identify which have a major effect and which can be deemed insignificant[18]. All possible levels of interaction are initially investigated, with insignificant higher-orders removed until a minimized model exists. This model can then be mathematically analysed by a generalization such as the General Linear Model (GLM) to give the percentage contribution (ϵ) that each parameter, including error, delivers to the overall system. The DoE was set up in Minitab as a two-level, four input, full factorial DoE with five repeats, giving 80 data points ($5\text{x}2^4$). Parameters were as follows:

- Level of Battery to increase Queue Length for Core Activation = 30/60
- Kick Out of non-priority jobs for priority jobs = Off/On
- λ=10/50
- μ=10/50

These results were analysed to determine key parameters for the GLM to determine the contributions each factor gives to the overall effect. These factors can allow further testing to give ideal results and an optimal scheduler design for the Heuristic Environment.

III. RESULTS

For the simulations undertaken, Table I shows the outcome for tests conducted where $\lambda = \mu$ and energy was considered. For this cycle, the Heuristic Algorithm outperforms both FCFS and Priority Queuing by a factor of 3, giving a statistically significant result. In cases where $\lambda \geq \mu$, the Heuristic Algorithm consistently outperforms its rivals and also gives significantly longer operating life in situations where $\lambda < \mu$ due to the dynamic deactivation of superfluous cores. This result is shown for other values of $\frac{\lambda}{\mu}$ in Fig 2, demonstrating that the Heuristic Scheduler outperforms both FCFS and Priority Scheduling for a range of values. No detriment to lifetime occurs with an increase in load rate, as proved by a regression test on the Heuristic Algorithm results (P=0.507, therefore no correlation between job rate and clock cycles completed).

When tests were conducted with single and dual core architectures the results, seen in Fig 3, show the Heuristic Algorithm continued to outperform both FCFS and Priority Queuing. With only one core active, Heuristic methods give

 ISBN: 978-1-61208-308-7

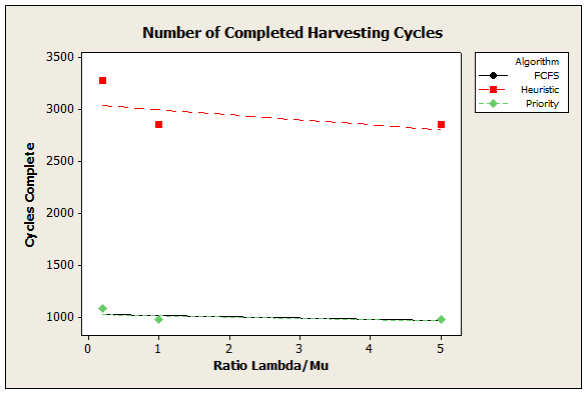


Fig. 2. Scatterplot showing Clock Cycles Complete for increasing values of lambda over mu

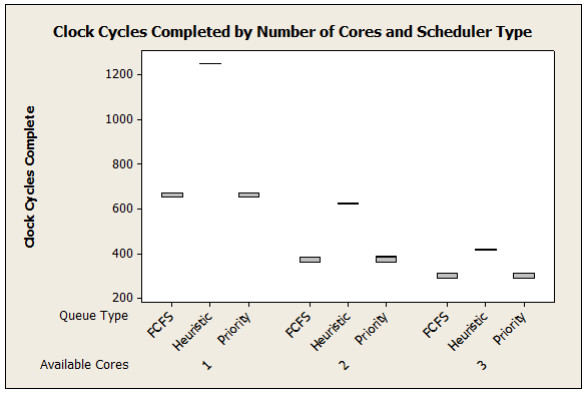


Fig. 3. Boxplot of Microprocessor Life for varying number of cores

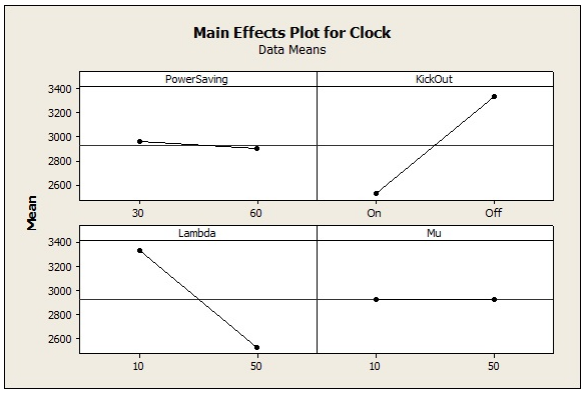


Fig. 4. Main Effects Plot of DoE

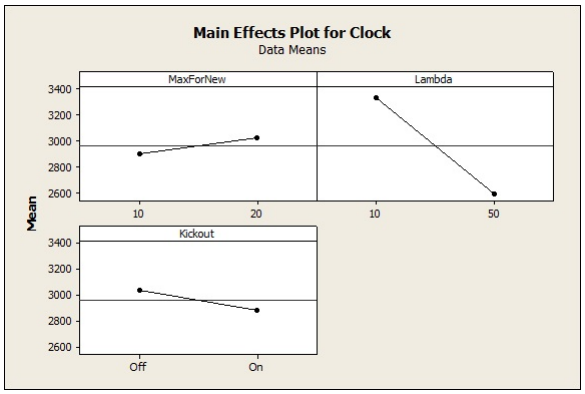


Fig. 5. Main Effects Plot showing additional parameters to Fig 4

a factor of two increase in operational lifetime due to the dynamic management of cores and their deactivation during light loading — similar to the sleep mode present in many modern microprocessors. As the core count increases, the effectiveness of the Heuristic Management can still be seen, giving improvements of 61 % for two cores and 35 % for three. This reduction in effectiveness is due to the increased dynamic and leakage power consumption present for a greater number of cores.

TABLE I
ONE-WAY ANOVA RESULTS FOR NUMBER OF CLOCK CYCLES COMPLETED COMPARED TO QUEUING METHOD WHEN $\lambda = \mu$

Queueing Method	Mean	Standard Deviation	P-Value	Conclusion
FCFS	971.00	0.25	0.00	Heuristic Outperforms
Priority Queue	970.96	0.32		
Heuristic	2848.79	0.54		

Following analysis of the DoE, Fig 4 shows an increase in Lambda, as well as deactivation of the Kick Out function have a significant effect on the number of successful clock cycles completed. Further analysis, shown in Fig 5, reveals that increasing the value of $MaxQueueLength$ in Algorithm 1 gives some increase to operational life and concurs with Fig 4 that being able to remove tasks from processors to allow execution of priority jobs (the "kickout" function) reduces operational life of the system – as this will increase the utilisation of processors and thus affect energy consumption. Due to this, a second analysis only looking at priority jobs was undertaken and can be seen in Fig 6. For this, deactivating Kick Out can be seen to have a large effect on missed priority jobs as these are now made to wait until a core has completed execution, rather than allowing the job immediate access.

The results from these experiments were placed into a GLM within Minitab to see determine main contributions to the

TABLE II
EPSILON SCORES FOR GENERAL LINEAR MODEL FOLLOWING DOE

Source	P-Value	ϵ
Power Saving	0.000	0.152
Kick Out	0.000	33.162
Lambda	0.000	33.097
Mu	0.756	0.000
Power Saving*Kick Out	0.000	0.153
Power Saving*Lambda	0.000	0.156
Power Saving*Mu	1.00	0.000
Kick Out*Lambda	0.000	33.125
Kick Out*Mu	0.022	0.000
Lambda*Mu	0.165	0.000
Power Saving*Kick Out*Lambda	0.000	0.155
Error		0.000
Total		100.0

 ISBN: 978-1-61208-308-7

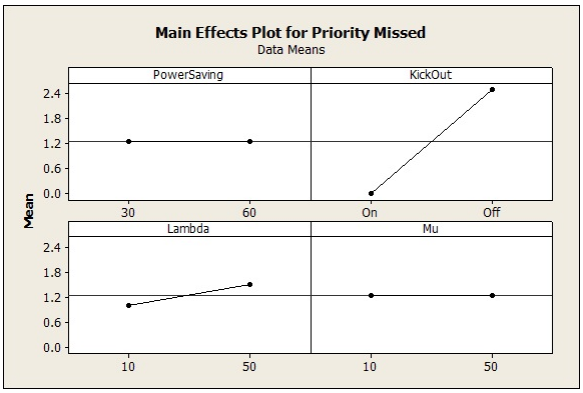


Fig. 6. Main Effects Plot analysing Priority Jobs only

microprocessor lifetime. The results for this, shown in Table II clearly show the contribution the kickout function and lambda have on the system, accounting for 99 percent of the effect seen. For these runs, error can be seen to be zero, due to the level of non-determinism afforded by the simulation being low. For repeat runs on a real microprocessor, this value would be expected to rise significantly.

IV. Conclusions

This work shows that under low loads ($\lambda < \mu$), a first come first served scheduler is capable of managing all jobs easily. With only one core active, FCFS gives a high value of maximum wait time compared to average wait time. Therefore this would not be suitable for a system with priority jobs. With two cores active however, this wait time reduces dramatically and priority queuing may not be required. However, when $\lambda > \mu$, priority queuing and multiple active cores becomes a feasible way to manage jobs.

The heuristic algorithm presented provides the flexibility of both strategies, combined with dynamic core management and therefore increased energy efficiency. By altering the queue length required to activate a new core with respect to energy available the algorithm, this extra non-deterministic aspect can be managed and the quality of service for an end user maintained.

When $\lambda \gg \mu$, a drop in performance for all scheduling methods takes place. As the arrival rate is so much larger than the service rate, the queue grows exponentially; meaning processor utilisation is always at 100 %. Due to this, no dynamic core management can take place and all schedulers perform at a comparable rate. Within these simulations, this leads to FCFS and Priority Queuing outperforming the Heuristic Algorithm, as the model designed considered the extra complexity required and increased leakage power accordingly — causing lifetime for the Heuristic Algorithm to be reduced.

While this system has only been tested up to three cores, it is thought that the algorithm would feasibly cope with a larger number of available devices in its current design. Since the scheduler operates as an overseer for all devices, it simply places jobs onto the first core it finds available; or activates a core if required. A limitation for this is the simulation work required to determine suitable values for $MaxQueueLength$ and the Energy at which to alter this. This work also currently presumes a homogeneous layout, but it is expected that heterogeneous microprocessors (where cores of different design are placed on the same silicon) will become more commonplace in the near future.

While this work only addresses a generic system with general job types, some further tests with real-time jobs have been conducted in a single-core variant of this work [19] and found to give an improvement in reliability over standard schedulers such as FCFS. This work developed an automotive Electronic Control Unit (ECU) within the MATLAB environment and had multiple priority levels for jobs. This further validates the work presented in this paper and shows the use of Heuristics in practical environments can give operational benefits.

V. Future Work

This paper proves the concept of a heuristic algorithm is viable and can dynamically manage a multi-core stored energy environment with minimal complexity. Key factors in its management have been identified and initially tested through simulation, which has allowed rapid testing of the many permutations and validation of the concept. The progression of this work is to conduct a Design of Experiments (DoE) to optimize the algorithm through determining each input variables overall contribution to both wait time and system reliability. DoEs have been used in previous work to great effect in identifying key items within an energy harvesting environment [20].

Owing to computational restrictions, only architectures up to three cores were investigated in this work. With the number of cores predicted to reach more than 300 by 2020 [21], tests of massive many-core layouts would be necessary to prove the versatility and usefulness of the Heuristic Algorithm against other more-advanced scheduling paradigms. While MATLAB is a useful tool for performing this, the runtimes for a 300 core simulation on a desktop computer would be excessive. One solution to this would be the use of a grid computer such as HTCondor to run the Monte Carlo simulation across a distributed system; thus reducing the test time and allowing validation of this concept on a massive many-core architecture.

As scalability may be an issue for heuristic algorithms, investigations into the use of pruning and filtering techniques are currently on going. These would give a hybrid approach, where certain situations would be managed by a standard schedule design, saving the use of heuristics for intensive or high-difficulty cases. Through this method, it is thought that the heuristic design could be preserved and used on more complicated systems without the need for redesign at each iteration.

While conceptually the design can be construed as sound, tests in a practical multi-core architecture will determine whether the algorithm works practically. Investigation for a suitable real-world processor and development kit are underway, with plans to port the algorithm into the system kernel

 ISBN: 978-1-61208-308-7

and perform tests against established schedulers including FCFS.

REFERENCES

[1] W. H. Wolf, *Modern VLSI design : systems on silicon*, 2nd ed. Upper Saddle River, NJ: Prentice Hall PTR, 1998.

[2] A. Bartolini, M. Cacciari, A. Cellai, M. Morelli, and A. Tilli, "Fault Tollerant Thermal Management for High-Performance Multicores," in *Workshop on Micro Power Management for Macro Systems on Chip as part of DATE 11*, D. Marculescu and S. Lesecq, Eds., Grenoble, France, 2011, accessed: Aug 5 2013. [Online]. Available: http://users.ece.cmu.edu/ dianam/uPM2SoC10/

[3] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, pp. 114–117, 1965.

[4] L. Torres, P. Benoit, G. Sassatelli, M. Robert, D. Puschini, and F. Clermidy, *An Introduction to Multi-Core System on Chip Trends and challenges*, 1st ed., M. Hubner and J. Becker, Eds. New York: Springer, 2011.

[5] A. S. Tanenbaum, *Modern Operating Systems*, 3rd ed. Upper Saddle River, N.J.: Pearson/Addison Wesley, 2009.

[6] J. Von Neumann and O. Morgenstern, *Theory of games and economic behavior*, 60th ed. Princeton, N.J. ; Woodstock: Princeton University Press, 2007.

[7] J. F. Nash, "Non-Cooperative Games," *The Annals of Mathematics*, vol. 54, no. 2, pp. 286–295, 1951.

[8] B. Foo and M. van der Schaar, "A Queuing Theoretic Approach to Processor Power Adaptation for Video Decoding Systems," *Signal Processing, IEEE Transactions on*, vol. 56, no. 1, pp. 378–392, 2008.

[9] A. Wierman, L. L. H. Andrew, and T. Ao, "Stochastic analysis of power-aware scheduling," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, 2008, pp. 1278–1283.

[10] S. Hsien-Po and M. van der Schaar, "Conjecture-based channel selection game for delay-sensitive users in multi-channel wireless networks," in *Game Theory for Networks, 2009. GameNets '09. International Conference on*, 2009, pp. 241–250.

[11] D. Gross, *Fundamentals of queueing theory*, 4th ed. Hoboken, N.J.: Wiley, 2008.

[12] Y. Nomura, M. Iwamoto, T. Yamanouchi, and M. Watanabe, "A heuristics guided scheduling framework for domains with complex conditions," *Proceedings Sixth International Conference on Tools with Artificial Intelligence. TAI 94*, pp. 752–755, 1994, accessed: 5 August 2013. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=346409

[13] I. Ahmad, "Using game theory for scheduling tasks on multi-core processors for simultaneous optimization of performance and energy," *2008 IEEE International Symposium on Parallel and Distributed Processing*, pp. 1–6, Apr. 2008, accessed: 5 August 2013. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4536420

[14] H. Cheng, "A High Efficient Task Scheduling Algorithm Based on Heterogeneous Multi-Core Processor," *2010 2nd International Workshop on Database Technology and Applications*, no. 3, pp. 1–4, Nov. 2010, accessed: 5 August 2013. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5659041

[15] D. Dal and N. Mansouri, "Power Optimization With Power Islands Synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 7, pp. 1025–1037, Jul. 2009, accessed: 5 August 2013. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5075810

[16] B. Zhao, H. Aydin, and D. Zhu, "Reliability-Aware Dynamic Voltage Scaling for Energy-Constrained Real-Time Embedded Systems," in *Computer Design, 2008. ICCD 2008. IEEE International Conference on*, vol. 546244, 2008, pp. 633–639.

[17] G. C. Buttazzo, *Hard real-time computing systems : predictable scheduling algorithms and applications*, 2nd ed. New York: Springer, 2005, accessed: 5 August 2013. [Online]. Available: http://www.loc.gov/catdir/toc/fy0613/2004058935.html

[18] F. W. Breyfogle Iii, *Implementing Six Sigma : Smarter Solutions Using Statistical Methods*, 2nd ed. Hoboken: John Wiley & Sons Inc., 2003.

[19] J. Docherty, A. Bystrov, and A. Yakovlev, "Simulation Testing of a Real-Time Heuristic Scheduler with Automotive Benchmarks," in *UK SIM 2013*, D. Al-dabass, Ed., Cambridge, UK, 2013.

[20] ——, "Identification of Key Energy Harvesting Parameters Through Monte Carlo Simulations," in *UK SIM 2012*, D. Al-dabass, Ed., vol. 0, Cambridge, UK, 2012, pp. 486–490, accessed: 5 August 2013. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/UKSim.2012.73

[21] S. Borkar, "Thousand Core Chips A Technology Perspective," in *Design Automation Conference*, 2007, pp. 746–749.

 ISBN: 978-1-61208-308-7

Reasoning on Concurrency: An Approach to Modeling and Verification of Java Thread-safe Objects

Franco Cicirelli, Libero Nigro, Francesco Pupo
Laboratorio di Ingegneria del Software
Dipartimento di Elettronica Informatica e Sistemistica
Università della Calabria - 87036 Rende (CS) – Italy
Email: f.cicirelli@deis.unical.it, {l.nigro,f.pupo}@unical.it

***Abstract*–Development of concurrent and time-dependent software systems is currently growing in its strategic importance due to the diffusion of powerful multi-core/many-core machines. To effectively cope with current and prospective concurrency demands, formal tools have to be used. A library of reusable UPPAAL timed automata was achieved, which enables a reasoning on concurrency. The library is tailored to Java. However, similar solutions could be also developed to work with other languages as well. This paper outlines library design and focuses on its exploitation for model-based prediction of the correctness of thread-safe Java objects.**

Keywords-Modeling and verification; concurrent systems; Java; thread-safe objects; model checking; UPPAAL.

I. INTRODUCTION

This work argues that to properly design and implement concurrent systems, the adoption of formal tools, which can support a *reasoning on concurrency* is mandatory.

This paper introduces the design of a UPPAAL library of timed automata [1-3], which improves preliminary experience described in [4]. Library design was mainly inspired by Java concurrency features. Common synchronization mechanisms such as semaphores and monitors, both classic and Java specific, are provided. On top of these mechanisms, new synchronizers, tailored to particular programming styles, can be built. The library enables modeling of a concurrent program according to implementation aspects, thus reducing the semantic gap which normally exists between a specification model and its vocabulary (e.g., atomic actions, broadcast synchronization, etc.) and a corresponding implementation. Analysis activities are based on exhaustive verification and model checking [5], [6].

The paper proposes an approach to modeling and verification (M&V) of Java thread-safe objects and illustrates it by practical examples.

The model-based prediction approach can be related to the work of Hamberg and Vaandrager [7] and to the known Finite State Processes (FSP)/Labeled Transition System Analyzer (LTSA) tool developed by Magee and Kramer [8]. With the first work our approach shares the use of the UPPAAL model checker. However, our library is characterized by its volition of being Java tailored. In addition, some common structures like semaphores appear to be more efficient. The second approach is based on the FSP process algebra. An FSP model is transformed into a labeled transition system to be analyzed by the LTSA tool. However, this approach does not allow, for instance, the expression of a FIFO policy (e.g., for awaking processes waiting on a semaphore). In addition, the use of a discrete time model can complicate the verification of complex models.

The paper is structured as follows. Section II outlines the developed UPPAAL. Section III proposes an approach for M&V of Java thread-safe objects. Section IV describes a more complex modeling example. Finally, an indication of research directions which deserve further work is given in the conclusion.

II. CONCURRENCY CONTROL IN UPPAAL

A library of UPPAAL template processes (i.e., timed automata –TA–) was developed, which provides such common concurrent structures as semaphores and monitors [9], both classic and Java specific. The following gives an outline of the library contents.

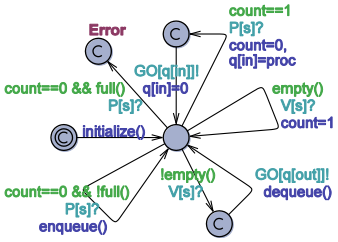


Figure 1. The BinarySemaphore automaton

Fig. 1 shows an automaton for classic binary semaphore. A similar construction exists for a general, counting semaphore. Classic P/V operations are implemented as unicast channel arrays P[.]/V[.] whose dimension mirrors the number of semaphores used in a model. A P operation on a semaphore s is expressed by raising a synchronization P[s]!. The requesting process is assumed to put into a global (meta) variable proc its unique process id at the time of P[s]!. Variable proc is used only during the atomic action of P[s]!, with the receiving semaphore which frees it immediately by storing the proc value in a local variable. A further channel array GO[.], whose dimension coincides with the number of processes in the model, is used for blocking the requesting process until the semaphore assigns the permit to the process. The use of GO is implicit in the operation P in a programming language, but in UPPAAL it serves the purpose of transforming a *strict rendezvous* (P[.]!) into an *extended rendezvous*, which terminates when the semaphore completes the handling of P[.]! and allows the requesting process to unblock. A V[s]! request never blocks the requesting process and normally does not require the proc mediation.

With respect to the realization proposed in [7], our semaphores use less variables thus favoring model

 ISBN: 978-1-61208-308-7

checking. For instance, the identity of the requesting process during a P operation, which finds a binary semaphore green (count==1) is temporarily stored in the surely empty internal queue of the semaphore.

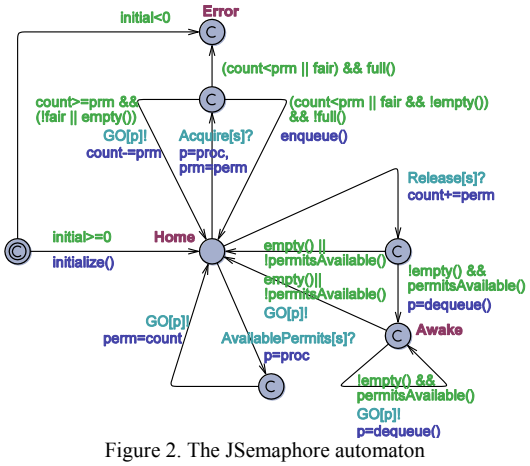


Figure 2. The JSemaphore automaton

A semaphore automaton (JSemaphore) directly based on a provided Java (java.util.concurrent.Semaphore) class is shown in Fig. 2. Differences from classic semaphores concern the possibility of acquiring/releasing atomically a number of permits greater than 1. In addition, a fair parameter can be used to request a FIFO behavior of acquire requests. This way, an acquire operation, in the case there are some waiting processes, puts the requesting process at the end of the semaphore queue even if permits are available.

JSemaphore relies on an interface consisting of channel arrays Acquire[.], Release[.], PermitsAvailable[.], GO[.], and exploits two global variables: proc and perm. The perm variable stores, at the time of an Acquire[s]! or Release[s]!, the number of involved permits, and contains the number of available permits of the semaphore just after a PermitsAvailable[s]! operation. A GO[p]? synchronization must follow an Acquire[s]! or a PermitsAvailable[s]! command. It is at the time of a GO[p]? unblocking operation, that perm is actually filled of the semaphore permits number.

It is worth noting that whereas a burst of release operations on a JSemaphore instance used as a mutex, will increase the permits number arbitrarily, in the case of a BinarySemaphore, a burst of V's can never augment the internal count beyond 1.

Although widely used, semaphores are often viewed as a low level concurrent abstraction mechanism, where a misuse of P/V operations can easily lead to a deadlock. Monitors (e.g., [9]), on the other hand, represent a higher level concurrent control structure, which naturally acts as a guardian of an abstract data type, e.g., encapsulated into a Java class. Monitors are a key for achieving thread-safe classes by offering control over mutual exclusion among class methods (synchronized blocks or critical sections of code) and suspension/signaling from within a critical section. Different kinds of monitors are defined in the literature, which are characterized by different programming styles and guarantees/obligations that are assigned to both processes and the control structure.

Java adopts, as a basic solution, the Lampson & Redell [10] monitor structure with broadcast signaling, where suspended processes in a synchronized block are responsible of re-checking a condition in a while loop to see, at each awaking, if it is necessary to coming back to waiting or the process can finally exit its waiting status. Broadcast signaling does not block the executing process. An awaken process has to compete in reacquiring the lock for it to actually resume execution.

Directly based on the built-in Java monitor structure is the JMonitor automaton presented in Fig. 3.

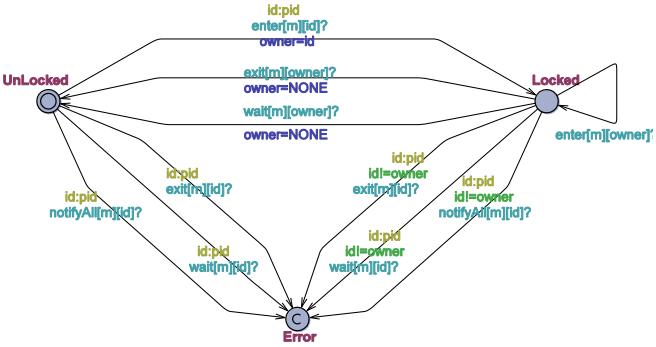


Figure 3. The JMonitor automaton

A monitor instance can be operated using such channel arrays as enter[mid][pid], exit[mid][pid], wait[mid][pid], notifyAll[mid][pid], which accommodate for the possible existence of multiple monitor instances in a model. Types mid and pid respectively are integer sub-ranges of unique identifiers of monitors and processes used in the model. For instance, enter[m][p]!/exit[m][p]! are used by a process p to explicitly enter/exit to/from a synchronized block based on monitor m. Similarly, wait[m][p]!/notifyAll[m][p]! serve respectively to suspend the requesting process p until its condition holds (in a while loop), and to awake all the processes suspended on monitor m.

Every Java object has an implicit lock which can be used as a monitor. The lock holds one implicit condition, whose meaning is established by the modeler/programmer. The lock object is associated with a *wait-set* where both entering processes which find the lock closed, and processes within a synchronized block (based on the lock object) whose condition prescribes waiting, are put (although the two kind of waiting processes are clearly distinguished to one another). Processes which are suspended for a wait operation can only be awaken by a notifyAll operation. The notifyAll operation does not free the lock. Other processes awake as the lock/monitor is up to be abandoned. In the proposed implementation, the wait-set is purposely realized implicitly. Processes requesting an enter are simply blocked if the monitor is already locked. Processes which execute wait are supposed to move into a location (see W1 and W2 in Fig. 6) from which they can only exit following a relevant notifyAll[.][.]? signal. Towards this, channels notifyAll[.][.] are declared as broadcast. Following a notifyAll signal, an awaken process has to compete for re-acquiring the lock (see edges exiting the W1 or W2 location in Fig. 6). Whereas this is implicit in the Java wait() method, it is explicit in the proposed modeling pattern, thus revealing a semantic issue.

 ISBN: 978-1-61208-308-7

The automaton in Fig. 3 maintains the identity of the monitor owner, which is used to realize reentrancy and to check for erroneous operations, which in Java correspond to raising an IllegalMonitorStateException, e.g., invoking a wait or notifyAll operation out of a synchronized block. The implicit realization of the wait-set complies with the Java specification and lets processes which try to enter the monitor and awaken processes to be handled non deterministically and thus without any privilege. The design makes it possible to implement a *timed wait*. In this case, from the wait location (now provided of a clock invariant) the process can also exit when the clock goes beyond a given time limit (*timeout*).

In reality, the Java built-in monitor also exposes a notify operation to awake *one*, although unspecific, process which is suspended in the wait-set. For generality reasons, the automaton in Fig. 3 only implements the notifyAll operation because, as discussed in [11], the use of notify can cause a *Lost-Wakeup-Problem*, i.e., a notify signal can be lost and the system enters a bad status.

On the basis of JMonitor, a monitor structure based on the Java Lock/Condition framework was also achieved, which allows to split the waiting processes among different conditions (*waiting rooms*) associated with a given lock. The signaling mechanism can be directed to a specific condition.

The library also includes some other classic monitors like the Hoare monitor [9], which in a case can be built on top of semaphores. The Hoare monitor owns a different signaling mechanism: when a process (*signaler*) changes the status of the data structure so that a (possibly) waiting process (*signalee*) on a condition can be awaken because the condition holds, control is immediately transferred to the signalee (together with the lock), which is thus the only process which can proceed. The signaler, on the other hand, is put to wait in an urgent queue from where it gets unblocked as soon as the monitor is up to become free.

A discussion about Lampson & Redell vs. Hoare monitors can be found in [9] where it is argued, besides any runtime implication (e.g., number of context switches), that Lampson & Redell monitor can be superior in the most general case.

III. An Approach to M&V of Thread-safe Objects

In the following, the proposed M&V approach is demonstrated by achieving a synchronizer based on two-way (or *rendezvous*) communications. The modeling example, which is original, can be used as a (partial) proof for programming a class like Exchanger<T> as provided in the java.util.concurrent package. The mechanism is intended to be used by two processes, which both play the sender/receiver roles. When the time arrives for a synchronization/communication, the earliest process which comes to the appointment awaits the partner. When the latest process arrives, processes exchange some information, then exit the synchronization thus returning to concurrent execution. In particular, the exchanger allows each process to send a message to the partner and to receive the message sent by the partner. Obviously, the mechanism can easily reproduce a CSP synchronous communication, when a partner is assigned the sender role and the other the receiver role.

An exchange synchronizer can be easily modeled by using the UPPAAL native features, as depicted in Fig. 4.

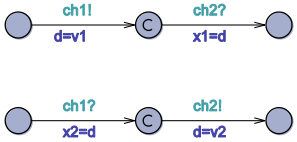


Figure 4. A native UPPAAL model for an exchanger

Fig. 4 assumes that each process transmits a local value v_i and receives from the partner an information to be stored in local variable x_i. A (meta) global variable d is used for the information exchange. Two unicast channels ch1 and ch2 are used where, for instance, ch1 can be declared as urgent. A committed location ensures an atomic exchange of information. Once the synchronization starts, on ch1, it is immediately followed (without a time passage) by a synchronization on ch2. Correctness of the model in Fig. 4 depends, among the other, on the fact that in a channel synchronization, the update (e.g., d=v1) of the sender (e.g., ch1!) is executed *before* the update (x2=d) of the receiver.

However, a native model like that in Fig. 4 cannot be immediately translated into Java, simply because the UPPAAL vocabulary of atomic actions, urgent channels, committed locations, etc. *is not* supported in the target language. The modeler/programmer is thus forced to intuitively achieve Java code by using the basic vocabulary of Java, e.g., synchronized blocks, wait/notifyAll, etc. However, the problem remains that an implementation *cannot* be proved to be a faithful representation of its specification model.

```
public interface Exchanger<T> {
   T exchange( T v );
}//Exchanger

public class ExchangerMJ<T> implements Exchanger<T>{
   private T d;
   private boolean partner = false, release = false;
   private Object m = new Object(); //lock/monitor object
   public T exchange( T v ){
      synchronized( m ){
         while( release ) //protection from a prompt re-enter
            try{ m.wait(); }catch( InterruptedException e ){}
         T x=null;
         if( !partner ){
            d = v; partner = true;
            while( partner ) //waiting for partner
               try{ m.wait(); }catch( InterruptedException e ){}
            x = d; release = false;
            m.notifyAll();
         }
         else{
            x = d; d = v; partner = false; release = true;
            m.notifyAll();
         }
         return x;
      }
   }//exchange
}//ExchangerMJ
```

Figure 5. A Java thread-safe class for the exchanger

To reduce the semantic gap existing between an UPPAAL model and a Java code, the model can embody implementation aspects. Stated in other terms, the model can mirror, through reverse-engineering, some Java code.

 ISBN: 978-1-61208-308-7

In Fig. 5, it is shown a Java code realizing the exchanger synchronizer. The Exchanger<T> interface exposes only the method exchange(v), which transmits v and receives the value sent by the partner. The ExchangerMJ<T> class implements, in a case, the Exchanger<T> interface in terms the of the native Java monitor.

Two waiting points exist in the exchange method in Fig. 5: one when the first process arrives and finds the partner is lacking (partner is false); another one for blocking a process, which finds a synchronization release in progress. When the latest process comes to the synchronization point, it finds partner=true, takes the transmitted data and sends its own information, then assigns false to partner, states that a release is up to commence, executes notifyAll and, finally, exits the synchronization block. Now, it is possible for the just exited process, to re-enter immediately the monitor whereas the last synchronization is still not finished. Apart for data overwriting problems, from Fig. 5 it is easy to see that a deadlock situation would occur.

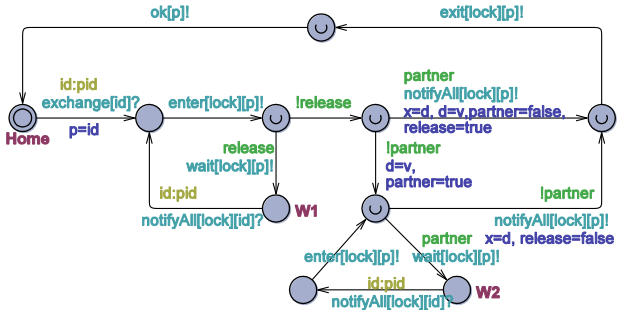


Figure 6. The Exchanger automaton based on JMonitor

Fig. 6 portrays a UPPAAL model for the ExchangerMJ class (integers are the exchanged data), which is based on the JMonitor automaton. Two arrays of channels are used: exchange[.] and ok[.], whose dimension is the number of processes. A process p requests an exchange through the operation exchange[p]! and then blocks on receiving an ok[p]? synchronization (see e.g., Fig. 7). Exchanger receives as a parameter the unique identifier of the lock object to be used internally for the synchronization. Since each participating process can wait at a different point in the control structure, two instances of the Exchanger automaton must be created, each serving a different process. Each instance links to the requisting process through a nondeterministic select (see the edge outgoing from Home in Fig. 6). In addition, information of the Exchanger (e.g., variables partner, release, etc.) are to be declared global to the model.

The use-pattern of Exchanger model in Fig. 6 is illustrated in Fig. 7 and Fig. 8, where a Producer sends the sequence 1, 2, 3 to a Consumer. Both processes are characterized by a process identifier (p) established at configuration time.

For proper behavior of the application, it is important that the consumer receives exactly the same data transmitted by the producer at each synchronization point. To check correctness of this behavior, the consumer model reaches the Error location as soon as it discovers an incorrect received data.

A system can be configured as indicated in Fig. 9.

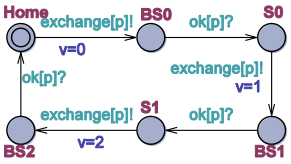


Figure 7. Producer automaton

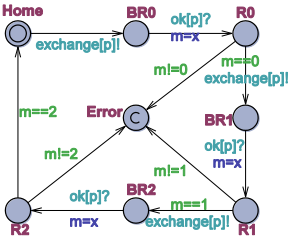


Figure 8. Consumer automaton

```
// Place template instantiations here.
ex0=Exchanger(LOCK);
ex1=Exchanger(LOCK);
//ex0=OptimisticExchanger(LOCK);
//ex1=OptimisticExchanger(LOCK);
prod=Producer(PROD);
cons=Consumer(CONS);
// List one or more processes to be composed into a system.
system JMonitor,ex0,ex1,prod,cons;
```

Figure 9. Producer/consumer system configuration

The following queries were issued to the UPPAAL model checker. The answers confirm all the queries are satisfied.

1) A[] !deadlock
2) A[] cons.R0 imply (prod.BS0 || prod.S0 || prod.BS1)
3) A[] cons.R1 imply (prod.BS1 || prod.S1 || prod.BS2)
4) A[] cons.R2 imply (prod.BS2 || prod.Home || prod.BS0)

Since there are no deadlocks, the consumer is guaranteed it never reaches the Error location. Query 2) ensures that when the consumer receives 0, the prod(ucer can't have completed the transmission of the next int. The producer completes the transmission of 1 when it enters the S1 location in Fig. 7. Similar considerations apply to queries 2) and 3).

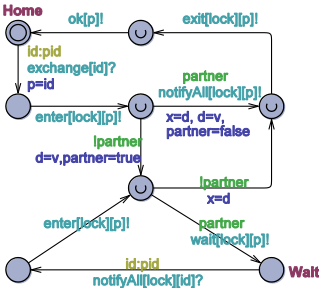


Figure 10. The OptimisticExchanger automaton

An optimistic variant of the Exchanger is shown in Fig. 10, which differs from Fig. 6 only because the first waiting point is eliminated.

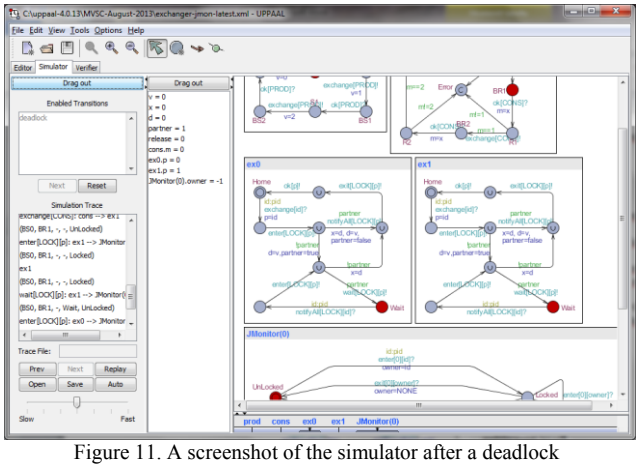


Figure 11. A screenshot of the simulator after a deadlock

By adjusting the system configuration in Fig. 9 so as to include two instances of the OptimisticExchanger template, it emerges that the first query is no longer satisfied. Asking the verifier to generate a diagnostic trace and opening it in the simulator, confirms the model is deadlocked (see Fig. 11).

For demonstration purposes, Fig. 12 shows an exchanger model based on semaphores. Two binary semaphores (with identifiers MUTEX and WAIT) are used: one as a mutex (initialized to 1), the other as a waiting room (initialized to 0).

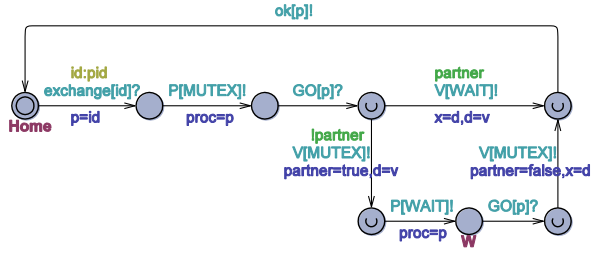


Figure 12. The ExchangerS automaton based on semaphores

Model in Fig. 12 rests on the "*pass the baton*" design pattern [12], i.e., when the latest process arrives (partner=true), it awakes the partner through a V[WAIT]! operation and then exits without releasing the mutex. As a consequence, the earliest process gets awaken *with* the mutex transferred to it. Therefore, a prompt re-enter is thus forbidden because mutex is occupied, and the release variable of Fig. 6 is useless.

Also, a system based on ExchangerS was configured and found correct by model checking.

As a final remark, except for the GO[.] synchronizations, the model in Fig. 12 directly maps on Java code.

IV. SECOND M&V EXAMPLE

The concept of a binary semaphore, corresponding to the UPPAAL model in Fig. 1, can be introduced in Java through a class as shown in Fig. 13. The realization relies on the language native monitor.

In order to check the correctness of the BinarySemaphore class, it was modeled in UPPAAL as depicted in Fig. 14, using the JMonitor automaton and the approach described in section III. Since the BinarySemaphoreMJ rests on JMonitor, it is assumed that also at the time of a V[.] operation the requesting process assigns its identifier to the global proc variable.

A notable difference between the automaton in Fig. 14 and the Java code in Fig. 13 concerns the realization of the linked waiting list, which in Fig. 14 is based on a bounded array managed as a FIFO queue. In addition, the toAwake integer variable is turned into a bounded int variable of UPPAAL, whose upper bound is qs+1 where qs is the queue size, established through a parameter of the BinarySemaphoreMJ template. As a consequence, when executing a burst of V's, it is useless to advance toAwake beyond this upper limit.

A key point of the model in Fig. 14 is the use of committed locations. The goal is to ensure that a P or V operation, once started, is conducted to a conclusion (i.e., re-entering the Home location or reaching the WaitTrue location) in an atomic way.

```
public interface Semaphore {
   void P();
   void V();
}//Semaphore

public class BinarySemaphore implements Semaphore{
   private int count, toAwake=0;
   private List<Thread> waitList=new LinkedList<Thread>();
   private Object m=new Object(); //lock-monitor object
   public BinarySemaphore ( int count ){
      if( count <0 || count >1 ) throw new IllegalArgumentException();
      this. count = count;
   }
   public void P(){
      synchronized( m ){
         if( count==0 ){
            waitList.add( Thread.currentThread() );//arrival order
            while( true ){
               try{ m.wait(); }catch( InterruptedException e ){}
               if( toAwake>0 &&
                  waitList.get(0)==Thread.currentThread() ){
                  toAwake--; waitList.remove(0);
                  if( toAwake>0 ){
                     if( waitList.size()>0 ) m.notifyAll();
                     else{ count =1; toAwake=0; }
                  }
                  break;
               }
            }//while
         }
         else count=0;
      }
   }//P
   public void V(){
      synchronized( m ){
         if( waitList.size()==0 ) count=1;
         else{ toAwake++; m.notifyAll(); }
      }
   }//V
}//BinarySemaphore
```

Figure 13. A FIFO BinarySemaphore Java class

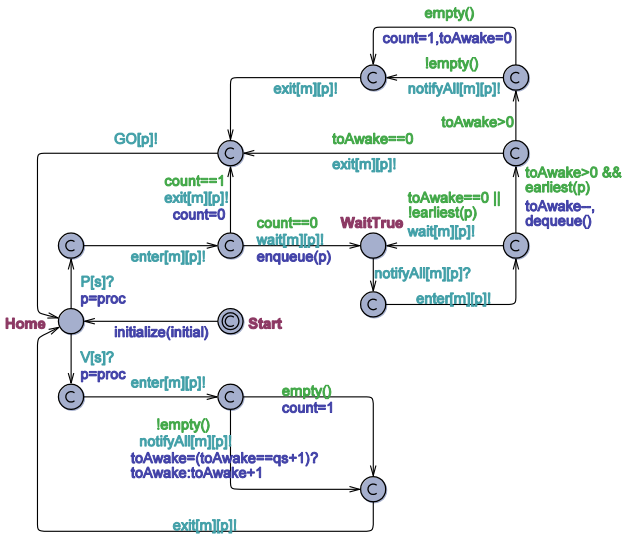


Figure 14. The BinarySemaphoreMJ automaton

An experimental verification frame was designed with the aim of comparing one instance (identifier S1) of BinarySemaphore in Fig. 1 and one instance (identifier S2) of BinarySemaphoreMJ in Fig. 14. Both instances receive a same sequence of P/V operations and the goal was to assess that both instances evolve exactly in the same way. Three process automata (see Fig. 15) were prepared: pProcess1, which acts on semaphore S1,

 ISBN: 978-1-61208-308-7

pProcess2 which operates on semaphore S2, and vProcess which uses both S1 and S2. Process instances receive as parameters: the unique process identifier (pid) p, and the names of used semaphores.

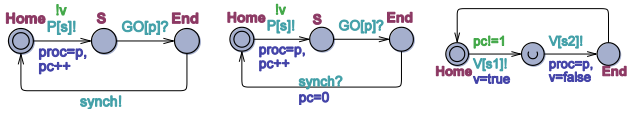


Figure 15. (a) pProcess1 (b) pProcess2 (c) vProcess automata

Since V operations are not blocking, one single instance of vProcess can be used to ensure that a V is actually issued to S1 *and* S2. Because a P operation can block the requesting process, one instance of pProcess1 and one instance of pProcess2 were used. A critical point in the process design was how to guarantee atomicity of the blocks {P[S1], P[S2]} and {V[S1], V[S2]}. Towards this, a global bool variable v is used, which is true if the V block is in execution. For the atomicity of the first block a global counter pc is employed, which is incremented each time a PO operation is launched. A V block can be started provided a P block is not in progress and similarly a P block can be started if no V block is in progress. Exiting a P block (see End location in Fig. 16 and Fig. 17) is ensured by a unicast channel signal synch, which is raised by a pProcess and received by other. At each synchronization over synch, pc is reset. Similarly, at the time of the second V of a V block, the variable v is reset. It should be noted that Fig. 15 (c) guarantees that a burst of V blocks can occur. Obviously, the semaphores S1 and S2 are supposed to be initialized to the same value.

The actual system configuration used for the verification experiments is shown in Fig. 16.

```
// Place template instantiations here.
bs=BinarySemaphore( S1, 0, PROC-2 );//id, init val, queue size
BS0=BinarySemaphoreMJ( S2, MON, 0 );//id, mon id, init val
BS1=BinarySemaphoreMJ( S2, MON, 0 );
p0=pProcess1( P0, S1 );
p1=pProcess2( P1, S2 );
v2=vProcess( V2, S1, S2 );
// List one or more processes to be composed into a system.
system JMonitor,bs,BS0,BS1,p0,p1,v2;
```

Figure 16. System configuration for the experimental frame

The following queries (all satisfied) were used for model checking:

1) A[] !deadlock
2) A[] p0.Home && p1.Home && v2.Home && BS0.Home && BS1.Home && bs.Home imply bs.count==count && empty() && bs.empty()
3) E<> p0.Home && p1.Home && v2.Home && BS0.Home && BS1.Home && bs.Home && count==0 && count==bs.count
4) E<> p0.Home && p1.Home && v2.Home && BS0.Home && BS1.Home && bs.Home && count==1 && count==bs.count
5) A[] p0.S && p1.S && (BS0.WaitTrue||BS1.WaitTrue) imply count==0 && bs.count==count && size()==1 && size()==bs.size() && first()==P1 && bs.first()==P0

Query 2) guarantees that when all automata are in their Home location, the semaphores have the same count value and their queues are both empty. Queries 3) and 4) show that in the same states of query 2), the semaphores can be both green or red. Query 5) verifies that when processes p0 and p1 are in the S location, i.e., they have both requested a P operation, in the case the BinarySemaphoreMJ is in the WaitTrue location, it effectively follows that both semaphores are red (count==0), their internal queues have the same size and in particular P0 is waiting in S1 and P2 is waiting in S2.

Meaning of queries from 2) to 5) ensures that after each *complete* execution of a block of P or V operations, the two semaphores have equivalent states.

Although the above described verification frame cannot replace a formal (weak) bisimulation proof of the two automata in Fig. 1 and Fig. 14, it provides important information about the correct behavior of BinarySemaphoreMJ and then of the Java thread-safe class in Fig. 13.

All the verification experiments were carried out on a Win 8, 12GB, Intel Core i7-3770K, 3.50GHz.

V. Conclusion And Future Work

The UPPAAL timed automata library and the M&V approach for concurrent systems proposed in this paper are useful in the practical case, and are under experimentation in an undergraduate course on systems programming. The solutions, although inspired by Java concurrency, can be adapted to work with other concurrent programming languages as well.

On-going and future work is geared at:

- Improving the supporting library of basic concurrent control structures and synchronizers.
- Extending the library in order to experiment with alternative but influencing concurrency schemes like the *software transactional memory* [11], which in the next future should be made available, e.g., in Java.

Acknowledgments

Authors thank Christian Nigro for his contribution to the development of the M&V approach described in this paper.

References

[1] R. Alur and D.L. Dill, “A theory of timed automata,” Theoretical Computer Science, vol. 126, no. 2, 1994, pp. 183-235.
[2] G. Behrmann, A. David, and K.G. Larsen, “A tutorial on UPPAAL,” In Formal Methods for the Design of Real-Time Systems, LNCS 3185, Springer, 2004, pp. 200-236.
[3] UPPAAL on-line, www.uppaal.org
[4] F. Cicirelli, L. Nigro, and F. Pupo, “Modelling and verification of concurrent programs using UPPAAL,” ECMS’2011, 2011, pp. 525-533.
[5] F. Cicirelli, A. Furfaro, and L. Nigro, “Model checking time-dependent system specifications using Time Stream Petri Nets and UPPAAL,” Applied Mathematics and Computation, vol. 218, no. 16, 2012, pp. 8160-8186, Elsevier.
[6] E.M. Clarke, O. Grumberg, and D.A. Peled, “Model Checking,” Cambridge, MA, MIT Press, 1999.
[7] R. Hamberg and F. Vaandrager, “Using model checkers in an introductory course on operating systems,” Operating System Review, vol. 42, no. 6, 2008, pp. 101-111.
[8] J. Magee and J. Kramer, “Concurrency – State models and Java programming,” John Wiley & Sons, Ltd., 2006.
[9] W. Stallings, “Operating Systems: Internals and design principles,” Prentice-Hall, 2005.
[10] B.W. Lampson and D.D. Redell, “Experience with processes and monitor in Mesa,” In Proc. of SOSP, 1979, pp. 43-44.
[11] M. Herlihy and N. Shavit, “The art of multiprocessor programming,” Elsevier, Revised version of First Edition, Morgan & Kaufmann Publishers, 2012.
[12] A.K. Reek, “Design patterns for semaphores,” ACM SIGCSE’04, 2004.

 ISBN: 978-1-61208-308-7

Monitoring and Modeling Web Server Performance: A Symbiotic Simulation Approach

Antonios Kogias, Mara Nikolaidou, and Dimosthenis Anagnostopoulos
Department of Informatics and Telematics
Harokopio University of Athens
Athens, Greece
coyas@hua.gr, mara@hua.gr, dimosthe@hua.gr.

***Abstract*—Existing approaches on web server simulation are often restricted, especially with the advent of the dynamic web. We propose a symbiotic approach for web server simulation, using a Faster than Real Time Simulation environment, compatible with the Dynamic Data Driven Applications Systems concept. The corresponding framework was implemented, consisting of: a measurement module, the FRT simulator, running concurrently with the web server, a controller that manages both the measuring process and the simulator, and a network level packet sniffer. Experimental results are presented along with open research issues.**

Keywords-modeling; web-server simulation; symbiotic simulation; faster-than-real-time simulation

I. INTRODUCTION

In symbiotic simulation [23], a simulation system and a physical system are closely associated with each other, in a potentially mutually beneficial relationship. The simulation system benefits from real-time measurements about the physical system provided by corresponding sensors. The physical system, on the other side, may benefit from the effects of decisions made by the simulation system. Operational decision making has hard real-time constraints and the manual evaluation of alternative decisions is difficult. Symbiotic simulation may alleviate this problem by automatically evaluating what-if scenarios within a reasonable period of time.

In Faster than Real Time Simulation (FRTS) [22][24], advancement of simulation time occurs faster than real world time. Making models run faster is the modeler's responsibility and certainly not a trivial task, since real time systems often have hard requirements for interacting with the human operator or other agents. Model evolution occurs faster than the real world and the experimentation results may be compared to the actual system and be used to improve the effectiveness of the simulation experiment. Incorporating into the model any occurring system changes is crucial for the reliability of the experiment; in FRTS, this happens in the process of remodeling, i.e., changing model specification in real time, as changes occur in the system.

Dynamic Data Driven Applications Systems (DDDAS) [21] is a concept of symbiotic relationship between application and measurement systems, wherein applications can accept and respond dynamically to new data, and reversely, the ability of application systems to dynamically control the measurement processes. The synergistic feedback control-loop between application simulations and measurements opens new domains in the capabilities of simulations with high potential pay-off, using sensors to produce large quantities of telemetry that are fed into simulations that model key quantities of interest. As data are processed, computational models are adjusted to best agree with known measurements. If properly done, this increases the predictive capability of the simulation system.

Web server modeling [6][10][14] and simulation [9][12], as well as http analysis [5][13] and web traffic modeling [11][15], while very active in the past, has received little contemporary attention, mainly due to the onslaught of the dynamic web and the inability of off-line simulations to use general models for the production of useful results. In this paper we propose the use of symbiotic simulation as an approach that could bring back the edge to the area, by enabling on-line simulations to use accurate and continually updated models, and produce useful insights about the real system's future (e.g., saturation, utilization, etc.) in real time.

The rest of the paper is organized as follows. In the second section, we present a brief review of web server simulation research, identify shortcomings and propose how to overcome them. In the third section, we present the proposed framework, and, in the fourth, the evaluation of our approach. We conclude at the fifth section.

II. WEB SERVER SIMULATION – OPEN ISSUES

The first published research in web server modeling and simulation [1] used a simple, high-level, open queuing network model (single server) and produced a theoretical upper bound on the serving capacity of Web servers. The single-server approach was also adopted in [3], where the model presented was an abstraction of the actions that occur at the session level layer, and all actions associated with the network layer were ignored, including specifics about individual TCP connections associated with requests (the web server was modeled as a single-server queue with single stream of Poisson arrivals). Colored Petri Nets (CPN) modeling was used in [4], where it was assumed that the fundamental service offered by a web server to web clients, is access to the documents stored therein. Only HTTP/1.0

 ISBN: 978-1-61208-308-7

was considered but it was noted that the CPN model could be easily modified to reflect HTTP/1.1. An end-to-end queuing model for the performance analysis of a web server was presented by Van Der Mei et al. [2], which described the impacts and interactions of the TCP subsystem, HTTP subsystem, I/O subsystem, and network, to predict the performance of web servers (in terms of end-to-end response time and effective throughput). This was a multi-server approach for static content only, although it was stated that the approach was valid for dynamic content as well. In the most recent web server performance analysis we found [7], it was noted that, considering the concurrent processing capability of modern web servers, it would be appropriate to consider them as multi-server systems. An M/G/m queuing model was presented, which was validated for deterministic and heavy-tailed workloads using experimentation. It was proposed that for most web servers, the capacity of the queue to hold requests when all the resources are busy was typically very large (to ensure that the probability of denying a request is very low); thus, queue size was assumed to be infinite.

Hereby, the restrictions we have identified in existing web server simulation approaches are discussed.

A. Complexity of dynamic content modeling

All models proposed so far have been specifically designed for, and tested with, static web server content, i.e., files of various types (HTML, JPEG, etc.) stored on a physical medium and accessed by the web server through the OS file system. Although a couple of approaches state that, since verified for static content, they are equally valid for dynamic content, there have been no reports of such successful attempts. Considering the simplifications already made to succeed in modeling solely static content, it is quite understandable that dynamic content proves very hard to model with acceptable degree of success. Apart from the already modeled response transmission time (with whatever complexity the existing static content models have established), there are other factors for which very little is (or can be) known, e.g., script engine (architecture, version, implementation platform, OS of availability), database engine (connection, efficiency, inter-networking factors, hardware parameters), quality and efficiency of the code that implements the dynamic application, etc. Apparently, the generality of static content approaches is of necessity lost, and a separate model must exist for each web application at a particular OS, network and hardware setup, at a specific point in time. Therefore, we decided to use a higher level modeling approach, focusing on the web server as a request processor. Dynamic applications, especially those that connect to databases, spend much more of their processing time retrieving data than preparing and sending the response to the client through the network. For dynamic content, what static content models are simulating is probably no longer the deciding factor for web server load.

B. No information about lost/denied requests

Content (i.e., TCP packets) gets lost during HTTP interactions over the internet all the time. As a rule, this is attributed to network congestion; however, another source is possible: web server overload. Web requests are processed from at least two queues: the TCP connection queue, where "socket-open" requests are gathered by the OS, and the HTTP request queue, where each accepted TCP connection waits until an http-thread (or process) becomes available to read, process the incoming request and send back the response. These queues, especially in today's computer systems where RAM is cheap, are typically very large – but definitely not infinite, although in all related work, they have been considered as such. Dropped requests never leave a trace in the web server's access log files, although they do use system resources; therefore, they should be modeled.

C. HTTP/TCP-specific end-to-end modeling

Most previous approaches have used the simplification that service time (server processing plus network I/O) is strongly related to the size of the HTTP response (in bytes), an approach that admittedly worked well for proposing improvements for the HTTP, and sometimes TCP as well, protocol. The models developed are very detailed and they lack the necessary simplicity for real life use – mainly because some, of their many, parameters are dynamic or impossible to measure, but also because the simulation running time becomes too long for effective use in real-time (or faster) setups. Therefore, a simpler, more abstract, service-based, and server-oriented approach is called for.

III. PROPOSED FRTS FRAMEWORK

The proposed approach does not deal with the TCP (and lower) subsystems, focusing instead at the HTTP layer and above; it measures performance and updates continually running simulations, which try to mirror the real system and predict its state in the future. It utilizes a simple web server model, combining an appropriate level of detail and faster-than-real-time execution speed in multiple replications within hard time constraints. It consists of: a) a measurement module, b) the faster than real time simulator, c) a controller that manages the simulator, acquires measurements and produces output, and d) a network level packet sniffer. The architecture of the proposed framework is depicted in Figure 1.

A. Measurement

The sniffer software (**Wireshark 1.6.0** [19]) runs on the same hardware with the web server and concurrently provides feedback on the network flows that reach it. Example settings are shown in Figure 2. The web server (**Tomcat 7.0.11** [16]) is the real system under test, which serves HTTP requests coming from web clients. The web server's access log has been formatted accordingly (using the "Valve" capability [18]) to facilitate easiness and speed of reading, as shown in Figure 3.

The **pattern** signifies that the web server logs (for each completed job) the time taken to process the request (**%D**) and the bytes sent including HTTP headers (**%B**).

 ISBN: 978-1-61208-308-7

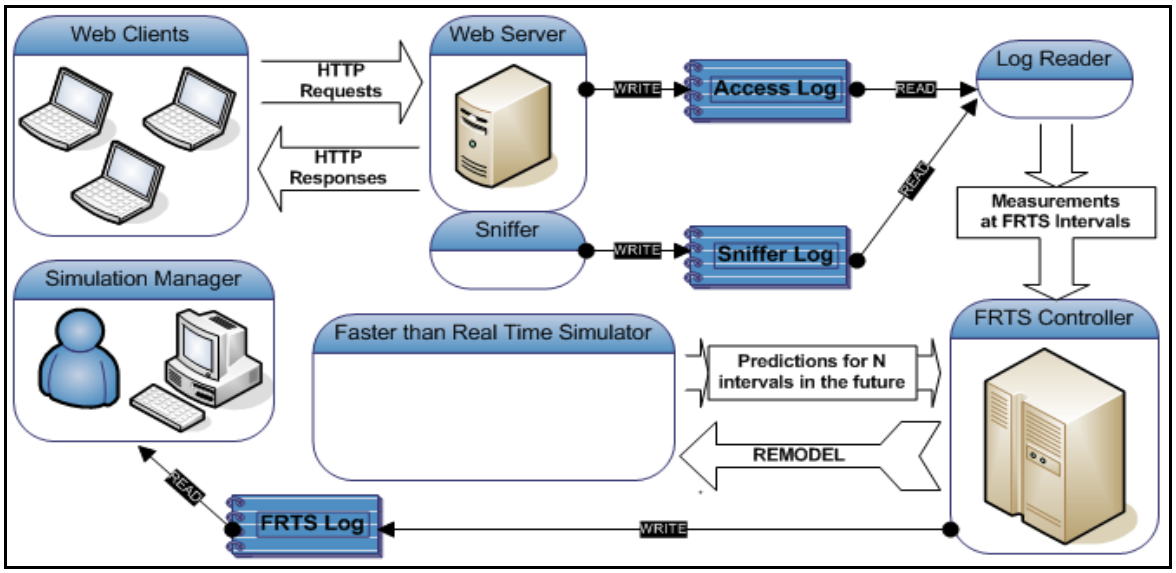


Figure 1. Components and data flow of the proposed framework

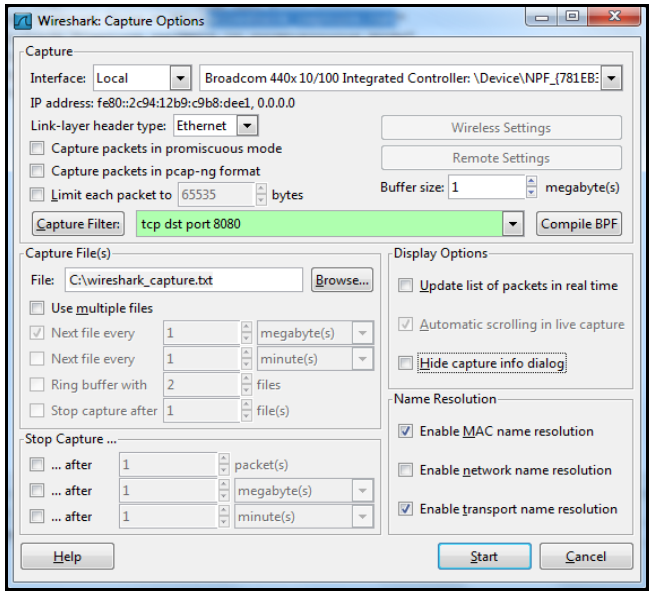


Figure 2. Wireshark capture options just before the Start button is pressed

```
<Valve
   className="org.apache.catalina.valves.
      AccessLogValve"
   directory="logs"
   prefix="localhost_access_log."
   suffix=".txt"
   pattern="%H %p %m %D %s %B %a %t
      &amp;quot;%r&amp;quot;"
   resolveHosts="false"   />
```

Figure 3. Tomcat's Valve configuration (conf/server.xml)

The web server's access log and the sniffer's capture file (i.e., sniffer log) constitute the measurement logs being used by the controller. The log reader component reads the new data entered in fixed time intervals, and provides the number of incoming HTTP ("GET") requests identified by the sniffer, the service time mean of the HTTP responses served by the server, and also the mean and deviation of the size of the HTTP responses.

Measurements are used by the FRTS Controller component to decide whether the simulation is still accurately depicting the real system or it has deviated due to real conditions changing. All the basic model parameters (setting the "Connector" section in the "server.xml" file [17] as seen in Figure 4) cannot be changed without web server restart (which would also necessitate a simulation restart), thus are only read once.

```
<Connector
   port="8080"
   protocol="HTTP/1.1"
   connectionTimeout="20000"
   redirectPort="8443"
   maxThreads="1"
   acceptCount="1"
   maxConnections="1"
   maxKeepAliveRequests="1"   />
```

Figure 4. Tomcat's Connector configuration (conf/server.xml)

The most significant measurement is **maxKeepAliveRequests**, which defines the maximum number of HTTP requests that can be pipelined until the connection is closed by the server. Setting this attribute to 1 will disable HTTP/1.0 keep-alive, as well as HTTP/1.1 keep-alive and pipelining. This property (when set to 1) makes the M/M/n model applicable to the web server, in a more abstract (HTTP and above) view of the web serving process without dealing with the TCP specifics. The property **maxThreads** signifies the maximum number of request processing threads to be created, i.e., the maximum number of simultaneous requests that can be handled. The property **acceptCount** signifies the maximum queue length for incoming connection requests when all possible request

 ISBN: 978-1-61208-308-7

processing threads are in use; any requests received when the queue is full will be refused. The property **maxConnections** sets the maximum number of connections that the server will accept and process at any given time; when this number has been reached, the server will not accept any more connections until the number of connections reach below this value – the operating system may still accept connections based on the **acceptCount** setting though. This is a property of slightly lower level than http (i.e., tcp and socket layer), which we set to **acceptCount + maxThreads**.

The controller compares the following four measurements with the predictions provided by the simulation for the past interval:

- Number of incoming requests
- Number of serviced requests (completed responses)
- Ratio of serviced to incoming requests
- Average size of responses

All of the predictions must be within the acceptability threshold of the measurements, provided as initial simulation parameter; if they were not, remodeling occurs: all available predictions are thrown away and the simulator is restarted with the latest measurements as parameters. The FRTS Controller runs on a separate thread, where it alternates between processing and sleeping the designated interval, taking the steps shown in Figure 5.

```
get predictions
read the sniffer log:
   measure arrivals/requests
read the access log:
   measure completed/serviced jobs
   measure mean servicing time
   measure average response size
get predictions from the simulator
compare predictions with measurements
if any of comparisons fails, then remodel:
   throw away all existing predictions
   incorporate latest measurements
   restart simulator
```

Figure 5. FRTS Controller process

B. Modeling and Simulation

The model we use in our simulation was implemented according to the Discrete Event Simulation (DES) paradigm [20] and is a simple queuing model with a single shared finite FIFO queue and a number of servers (n) that service jobs waiting in queue and cannot be idle unless the queue is empty (M/M/n in queuing theory because both arrivals and service times are memoryless, i.e., exponential). The queue models the http-queue that the web server has, where incoming http-requests are held waiting until an http-thread becomes available to process and produce/transmit the http-response. Servers correspond to available http-threads that the web server has and are used to process incoming http-requests and transmit http-responses. The controller manages a number of faster than real time simulators, each running an M/M/n model. At real time intervals of the equal duration to the simulation intervals, each simulator momentarily pauses to gather statistics for that particular predicted interval, and then continues simulating from the exact moment it had paused. This way, the controller is able to provide predictions about the values of interest (requests, responses, size) for the specified intervals in the future.

The interarrival time distribution is considered exponential and its mean (beta) is computed from the sniffer's capture file measurements of the amount of incoming http requests in the latest measurement interval. The service time distribution is considered exponential and its mean (beta) is from the web server's web access log measurements for each successfully processed http request and subsequent http response. The response size distribution has been chosen to be Gaussian/Normal. The mean and deviation of the response sizes is computed from the web access log and passed to the simulation. The measure of success is the frequency of remodeling events during its execution: the lower the better.

IV. EVALUATION

Our aim was to provide validation and verification of the framework for use with static content web servers in controlled (laboratory) conditions, so we did extensive experimentation with static content, controlling both the web resources dataset available, the test web client and the web server settings, as described below.

A. Datasets

There are four fixed-size file datasets, each consisting of randomly generated files of fixed (per dataset) size equal to the number in parenthesis in kilobytes:

- F(1): 10k different files of 1kb size
- F(10): 1k different files of 10kb size
- F(100): 100 different files of 100kb size
- F(1000): 10 different files of 1000kb size

The total volume of each dataset was set to be the same, 10 megabytes; they were used for fine-tuning the simulation setup and as the backbone for varied-size experimenting.

We used combinations of the F datasets for creating varied-size datasets, by merging the various F datasets in all possible combinations; thus creating skewed probabilities of response size, with the intent of proving that the simulation setup is versatile enough to cope with such traffic:

- V(A): F(1) + F(10)
- V(B): F(1) + F(100)
- V(C): F(1) + F(1000)
- V(D):F(10) + F(100)
- V(E): F(10) + F(1000)
- V(F): F(100) + F(1000)
- V(G): F(1) + F(10) + F(100)
- V(H): F(1) + F(100) + F(1000)
- V(I): F(1) + F(10) + F(1000)
- V(J): F(10) + F(100) + F(1000)
- V(K):F(1) + F(10) + F(100) + F(1000)
- V(L): 35x1kb + 50x10kb + 14x100kb + 1x1000kb

For example, the V(D) setup consists of merging the F(10) and the F(100) datasets; therefore, the probability of

 ISBN: 978-1-61208-308-7

requesting a 10 kilobytes file is ten times more than that of a 100 kilobytes file. V(L) is a dataset of special proportions, as it comprises of 35 files from F(1), 50 files from the F(10), 14 files from the F(100) and 1 file from the F(1000). It is an effort to represent the SPECweb96 benchmark [25], which has been used extensively for static content web server simulation in the past. The SPECweb96 workload defines four classes of files to get, based on the following file sizes: less than 1 KB, 1 to 10 KB, 10 to 100 KB, and 100 KB to 1 MB (there are several files in each class, with sizes distributed evenly through the range for that class). SPECweb96 directs 35 percent of its activity to the smallest class, 50 percent to the 1-to-10-KB class, 14 percent to the 10-to-100-KB class, and one percent to the largest files.

B. Simulation Setup

For the final experimentation setup, we picked 30 simulators as the best compromise between accuracy and performance (increasing their number did not significantly increase accuracy) and acceptance threshold of ±20% (i.e., system will not proceed to remodeling if predicted values are within 80% - 120% of measured values) over the four measurements monitored.

The monitoring interval was set at 30 sec, to provide for RAM conservation, emergent dynamic HTTP variability and modeling suitability. It was the lowest value that allowed for consistent simulation in our experiments; with values below 30 sec, the simulations had trouble converging. The prediction window was set at 10 intervals into the future (i.e., 5 minutes of real time); if a simulator reached that threshold of predictions, it went to 'sleep' to conserve system resources. Each experiment lasted at least 40 intervals (i.e., more than 20 minutes of real time), a long enough duration for interesting phenomena to emerge.

We used one multi-threaded web client to create the server workload with exponential inter-request rate and uniform random selection of file requested from all those available in each dataset (thus creating the skewed probabilities explained earlier).

C. Experimentation

The web server was setup to run with either **1** or **100** processing threads using a queue of either **1** or **100** pending requests. These values were of course mirrored in the FRTS simulators for accurate modeling. The web client was setup to create either **10** requests/sec or **100** requests/sec, values that proved during fine-tuning to be the thresholds for interesting behavior and implementation stress.

We run experiments with these eight different combinations over the four F datasets (32 experiments) and the twelve V datasets (96 experiments); **128** experiments in total. For each experiment we measured the percent of overall FRTS success, i.e., the ratio of intervals that predictions were within acceptance threshold (i.e., no remodeling) over the total intervals of the simulation run.

D. Results

The overall simulation setup ran quite smoothly, although FRTS implementation RAM issues lead four F(1000) and one V(F) experiments of high request rate (100 requests/sec) to early shutdown (marked as invalid).

We found that using the Normal distribution (Gaussian) for predicting response size was a poor choice because it went astray in most V datasets, causing remodelings that could otherwise have been avoided. Size prediction (**Z**) is traditionally important for predicting the response benchmark (**S**); in our model however those are disjoint. Therefore, remodelings due to **Z** alone were excluded from success ratio calculations. Success percentages presented below account only for **ASR** remodelings, considering **Z** remodelings as never occurred.

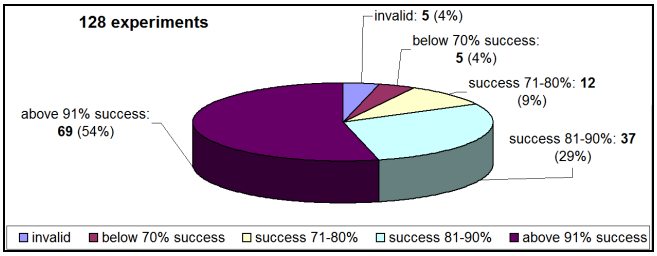


Figure 6. Results of all 128 experiments

All experiments showed greater than 65% success, with more than half of them in the 91-100% scale. In Figure 6, the results of all 128 experiments are shown in detail. The F datasets experiments were the less important (included mostly for sanity check). The V dataset experiments were considered more important; especially the V(L), in which success rates were consistently above 90%. The simulation was slightly more successful when request rate was low than high; it also lost some accuracy in high size datasets and those of great variability. However such failings were expected, given the simplicity of the model we used. In general, the simulation quality seemed unaffected by the web server's ability to cope with the load, providing quality predictions even in the interesting occasions that the web server could not cope with the load.

V. CONCLUSION AND FUTURE WORK

Research in the area of web server simulation is active and useful; however, traditional off-line simulations have trouble dealing with the onslaught of dynamic web and the lack or relevant generic models. The proposed framework is a novel symbiotic simulation approach in this direction, with the potential of breaking through the barrier of web server dynamic content modeling and simulation.

It is apparent that the framework is 'cold' restarted after each remodeling, but that cannot be helped as whatever state the simulator has reached must be considered invalid. We are currently working in incorporating distribution estimation, instead of simple means, into the framework. The model used is arguably very simple, and more complex versions, along with concurrent simulators of different models, should be developed. Profiling web traffic and

 ISBN: 978-1-61208-308-7

workload classes, delving deeper into the complexity of the web server, as well as the stability of the measure and remodel loop are also open approaches that we wish to investigate. Interesting future expansions include decision processes to modify the interval duration and self tuning some real system parameters.

After exploiting the above directions and completing testing of the proposed framework for static content web server simulation, research will focus solely on its applicability for dynamic content.

REFERENCES

[1] L. P. Slothouber, "A Model of Web Server Performance," 5th International Web Conference, 1996 (poster).

[2] R. D.Van Der Mei, R. Hariharan, and P. K. Reiser, "Web Server Performance Modeling," Springer Telecommunication Systems, Vol. 16, Issue 3-4, March 2001, pp. 361-378.

[3] A. C. Dalal and S. Jordan, "Improving User-Perceived Performance at a World Wide Web Server," Proc. IEEE Global Telecommunications Conference (GLOBECOM '01), IEEE, Vol. 4, pp. 2465-2469, 2001, doi:10.1109/GLOCOM.2001.966220.

[4] L. Wells, S. Christensen, L. M. Kristensen, and K. H. Mortensen, "Simulation Based Performance Analysis of Web Servers", Proc. 9th international Workshop on Petri Nets and Performance Models (PNPM'01), IEEE Computer Society, 2001, p. 59.

[5] C. D. Murta and G. N. Dutra, "Modeling HTTP Service Times," Proc. Global Telecommunications Conference (GLOBECOM '04), IEEE, Vol. 2, pp. 972-976, 2004, doi:10.1109/GLOCOM.2004.1378104.

[6] V. V. Panteleenko and V. W. Freeh, "Web Server Performance in a WAN Environment," Proc. 12th International Conference on Computer Communications and Networks (ICCCN 2003), pp. 364-369, 1997, doi:10.1109/ICCCN.2003.1284195.

[7] J. Lu, and S. S. Gokhale, "Web Server Performance Analysis," Proc. 6th international conference on Web engineering (ICWE '06), ACM, 2006, pp. 111-112, doi:10.1145/1145581.1145605.

[8] L. Li, R. X. Tian, B. Yang, B., and Z. G. Gao, Z. G, "A Model of Web Server's Performance-Power Relationship," Proc. IEEE International Conference on Communication Software and Networks (ICCSN '09), IEEE Computer Society 2009, pp. 260-264, doi:10.1109/ICCSN.2009.131.

[9] Y. Hu, A. Nanda, and Q. Yang, "Measurement, Analysis and Performance Improvement of the Apache Web Server," Proc. 1999 IEEE International 12th International Performance, Computing and Communications Conference, 1999, pp. 261-267, doi:10.1109/PCCC.1999.749447.

[10] A. E. Hassan and R. C. Holt, "A Reference Architecture for Web Servers," Proc. 7th Working Conference on Reverse Engineering (WCRE'00), IEEE Computer Society, 2000, pp. 150-160.

[11] M. S. Squillante, D. D. Yao, and L. Zhang, "Web Traffic Modeling and Web Server Performance Analysis," ACM SIGMETRICS Performance Evaluation Review, Vol. 27 Issue 3, ACM, Dec. 1999, pp. 24-27, doi:10.1145/340242.340323.

[12] S. Gokhale, U. Praphamontripong, A. Gokhale, and J. Gray, "Performance Analysis of an Asynchronous Web Server," Proc. 30th Annual International Computer Software and Applications Conference (COMPSAC '06), Vol. 02, IEEE Computer Society, 2006, pp. 22-28, doi:10.1109/COMPSAC.2006.148.

[13] J. Heidemann, K. Obraczka, and J. Touch, "Modeling the Performance of http Over Several Transport Protocols," IEEE/ACM Transactions on Networking (TON), Vol. 5 Issue 5, Oct. 1997, pp. 616-630, doi:10.1109/90.649564.

[14] J. C. Hu, I. Pyarali, and D. C. Schmidt, "Measuring the Impact of Event Dispatching and Concurrency Models on Web Server Performance Over High-speed Networks," Proc. Global Telecommunications Conference (GLOBECOM '97), IEEE, Vol. 3, pp. 1924-1931, 1997, doi:10.1109/GLOCOM.1997.644610.

[15] B. Liu and E. A. Fox, "Web Traffic Latency: Characteristics and Implications," Journal of Universal Computer Science, Vol. 4, No 9, pp. 763-778, 1998, doi:10.3217/jucs-004-09-0763.

[16] "Apache Tomcat 7," http://tomcat.apache.org, 22.08.2013.

[17] "The HTTP Connector," http://tomcat.apache.org/tomcat-7.0-doc/config/http.html, 22.08.2013.

[18] "The Valve Component," http://tomcat.apache.org/tomcat-7.0-doc/config/valve.html, 22.08.2013.

[19] "Wireshark," http://www.wireshark.org, 22.08.2013.

[20] A. M. Law and W. D. Kelton, "Simulation Modeling and Analysis," McGraw–Hill, 2000, pp. 6-57.

[21] C. C. Douglas, "Dynamic Data Driven Applications Systems – DDDAS 2008," Proc. 8th International Conference on Computational Science, Part III, Springer Lecture Notes in Computer Science, Vol. 5103, 2008, pp. 3-4.

[22] D. Anagnostopoulos, M. Nikolaidou, and P. Georgiadis, "A Conceptual Methodology for Conducting Faster Than Real Time Experiments," Transactions of the Society for Computer Simulation International, Vol. 16 Issue 2, pp. 70-77, June 1999.

[23] H. Aydt, S. J. Turner, W. Cai, and M. Y. H. Low, "Research Issues in Symbiotic Simulation," Proc. Winter Simulation Conference (WSC '09), 2009, pp. 1213-1222.

[24] D. Anagnostopoulos and M. Nikolaidou, "Executing a Minimum Number of Replications to Support the Reliability of FRTS Predictions," Proc.7th IEEE International Symposium on Distributed Simulation and Real-Time Applications (DS-RT '03), IEEE Computer Society, 2003, p. 138.

[25] "Answers to Common Questions About the SPECweb96 Benchmark," http://www.spec.org/web96/web96q+a.html, 22.08.2013.

 ISBN: 978-1-61208-308-7

A Flexible Analytic Model for a Dynamic Task-Scheduling Unit for Heterogeneous MPSoCs

Oliver Arnold, Benedikt Noethen, and Gerhard Fettweis
Vodafone Chair Mobile Communications Systems
Technische Universität Dresden
Dresden, Germany
{oliver.arnold, benedikt.noethen, fettweis}@tu-dresden.de

Abstract— **In this paper, a heterogeneous Multiprocessor System-on-Chip (MPSoC), controlled by a dedicated task scheduling unit, is presented. This unit, known as CoreManager, is responsible for dynamic data-dependency checking, task scheduling, processing element allocation and data-transfer management. Three different CoreManager approaches are analyzed and compared. An analytical model is derived for each CoreManager implementation. The configuration parameters for the models are determined through system analysis. For this purpose, a tool flow has been developed to build the MPSoC and generate data traces. For the benchmarks employed, the relative error of the analytical model was shown to be lower than 6.3 % on component and 6.9 % on system level compared to the measurements.**

Keywords-Heterogeneous MPSoC, Dynamic Task Scheduling, CoreManager, Analytical Model

I. INTRODUCTION

Multiprocessor System-on-Chips (MPSoCs) are composed of several types and numbers of processing elements (PEs) and allow increasing performance and energy efficiency. In order to cope with the stringent performance-efficiency requirements, architectures exploiting parallelism and data locality both at system and core level [1] are required. Even though data-level and instruction-level parallelism within the PEs is essential, the main focus of this work is in the functional, i.e., task-level parallelism, based on the data flow model [2].

Increasing the system complexity in terms of application parallelism and number and types of resources may lead to a dramatic increase of system management costs, thus causing performance degradation. For this reason, the efficient implementation of the management unit becomes a major issue in system design. Therefore, an analytical model is necessary to predict and analyze the runtime behavior of the management unit and the heterogeneous system.

This work compares the performance and capabilities of a dedicated task scheduling unit, called CoreManager. Three different implementation approaches are regarded: a RISC-based solution (CM-RISC), an approach with Very Long Instruction Words (CM-VLIW) and an implementation based on an extended instruction set architecture (CM-EIS). A flexible analytical model has been derived for each implementation approach. Furthermore, a tool flow has been developed to build a heterogeneous MPSoC and to generate data traces. The configuration parameters for the models have been analytically derived and the obtained results compared to the measurements.

Some examples of heterogeneous hardware platforms are the Cell Broadband Engine [3] and Sandbridge SB3011 SDR platform [4]. The Tomahawk MPSoC was developed to execute applications from the multimedia as well as the signal processing domain [5]. It includes a dedicated task scheduling unit. In [6], a comparison between a software and a hardware scheduling approach is presented. The programming model used in this work is similar to CellSs [7]. Further programming models are, e.g., Cilk [8], Sequoia [9], and Ct [10].

The extension of the instruction set of standard processors is available in many areas [11][12]. In this work, a RISC core is extended by several newly introduced instructions to improve task scheduling performance as well as energy consumption. A similar approach was presented in [13].

According to the taxonomy given in [14], the used dynamic task scheduling is centralized and applies complete information exchange to schedule aperiodic tasks. Complete information exchange refers to the collection of events from all processing elements. The platform used in this work can be understood as a distributed system due to the separate address spaces of the processing elements [15].

The remainder of the paper is organized as follows. In section II, the hardware system and the programming model are presented. In the following section, the tool flow is described. Section IV presents the components of the task scheduling unit, called CoreManager. It is analytically described in the next section. Section VI shows the results of the system. The parameters of the analytical models are presented. Furthermore, a comparison of the analytical model and the measurements is given.

II. SYSTEM MODEL

A. Hardware Model

A heterogeneous MPSoC is depicted in Fig. 1. It consists of several functional blocks, which are connected by a Network-on-Chip (NoC). A router is available for each system component, which is connected to its neighbors by point-to-point data links. The routers are responsible for

 ISBN: 978-1-61208-308-7

packet scheduling and arbitration. XY routing is applied. Further details about the integrated NoC can be found in [16].

The Application Processor (APP) is formed by a Tensilica 570t core and has 2-way set-associative instruction and data caches, each 16 Kbyte in size. It is placed next to an off-chip memory interface for fast data access. The data plane of the system is composed of several types and numbers of processing elements (PEs), which are controlled by the CoreManager. The CoreManager is responsible for task scheduling, PE allocation, and data transfer management. The CoreManager presents an interface which allows connecting to the application running on the APP.

Three off-chip global memories are included (MEM_0, MEM_1 and MEM_2), each one having 256 MB. Each PE has its own dedicated direct memory access controller (DMAC) to perform data transfers between the global memories and their local memories. Furthermore, data can be fetched from local memories of other PEs.

Two types of PEs are integrated in the system: a digital signal processor (DSP) and a RISC processor. For each type, ten processors are instantiated. In the proposed approach, a PE can solely operate on its local on-chip memory. No cache misses can occur. Task execution time is consequently deterministic, which leads to a better predictability at system level. PEs' instruction and data memory size is 32 Kbyte each. Prefetching of data is possible for the next two tasks, but must be explicitly annotated by the CoreManager. Similar to the PEs, the CoreManager solely works on local on-chip memories. Its instruction and data memory size are 32 Kbyte each. Data transfers to the local memories of the PEs and task execution can be performed concurrently. A clock frequency of 333 MHz is applied for all components.

B. *Programming Model*

The used programming model, called taskC, is based on tasks as a main entity [15]. A task is a collection of instructions which are atomically executed. In Fig. 2, a source code example is shown. For each task input and output, data transfers are specified with IN, OUT and INOUT operators. For each transfer, a pointer and a size are specified at runtime. 2-dimensional data transfers are supported. For example, in software defined radio systems, the data locations of a task are specified after the header is processed. No static data analysis is possible for these kinds of applications.

The task execution is not done by the APP itself. The APP only sends the task description, which is composed of the task name and the data information, to the CoreManager. In Fig. 2, two task descriptions are transferred, either taskType1 and taskType2 or taskType1 and taskType3. The APP is additionally responsible for evaluating control-code dependencies, e.g., the if-else clause in Fig. 2. Data-dependencies between tasks are evaluated by the CoreManager at runtime. The taskSync command is a barrier and synchronizes the APP and the data plane execution. After the APP returns from this function it is assured that all tasks are finished and all output transfers have been completed.

III. TOOL FLOW

A newly developed tool flow is used to specify the system configuration and to generate the simulation environment. An overview of all components is shown in Fig. 3. The hardware architecture is specified in a configuration file containing two parts. The first part is responsible for the system level. The second part specifies the capabilities of the CoreManager. By using the Tensilica Xtensa Processor Generator (XPG) the CoreManager as well as the PEs are created [17]. RTL code and suitable Compilers are generated as well. The InstGenerator and the TaskCompiler are responsible for the compilation of tasks and their extraction into a separate data array. The application itself is compiled with the Tensilica 570t Compiler. Binaries for the PE and the CoreManager are linked into the APP binary. These binaries are loaded at runtime to the corresponding cores.

Three types of hardware designs are generated: A Tensilica-based cycle-accurate simulation environment (XTSC), a FPGA prototype, and an ASIC prototype. The TaskVisualizer allows visualization of results. In particular, it shows task execution and data transfers. More information on the TaskVisualizer can be found in [15]. The CoreManager Profiler and the DebugVisualizer allow an offline and online analysis of the CoreManager. More information on these tools can be found in [18].

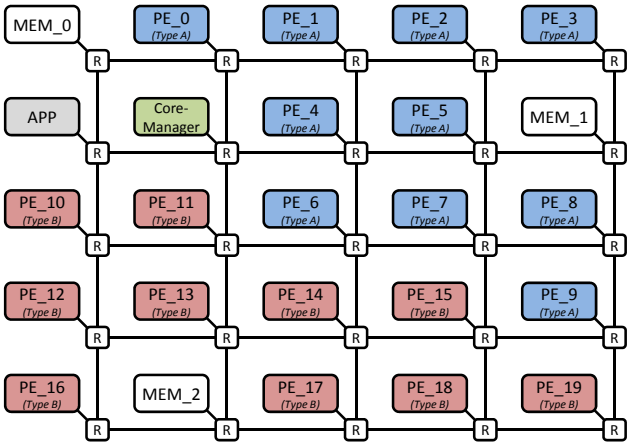


Figure 1. System Model: heterogeneous MPSoC

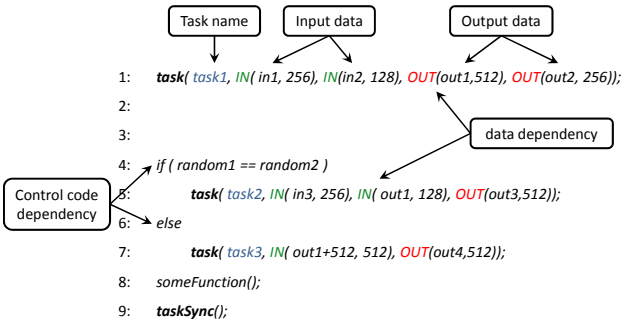


Figure 2. Programming model example

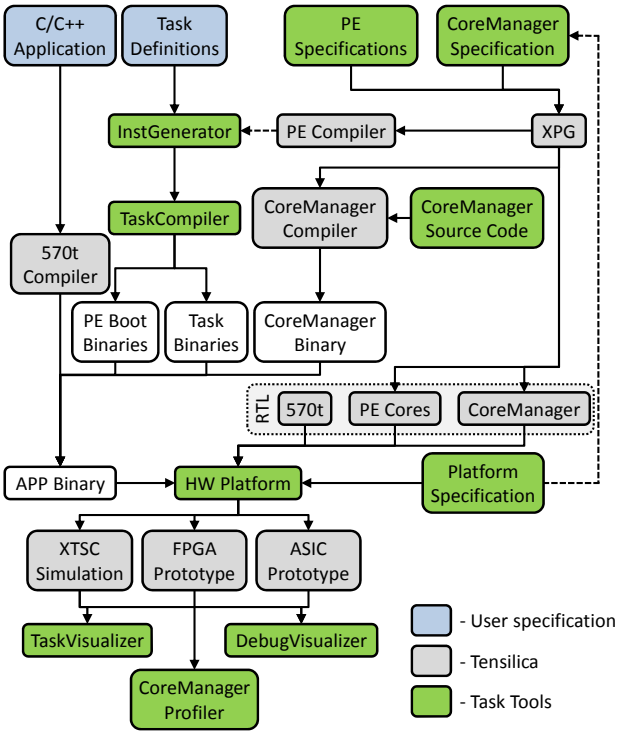


Figure 3. Tool flow

IV. CoreManager Structure and Behavior

The major components and the internal data flow of the CoreManager are depicted in Fig. 4. The operational sequence is as follows. Firstly, the APP retrieves the ID of an empty task slot by reading the CM_2_APP first-in first-out (FIFO) memory (step 1). Afterwards, the APP writes the task description (e.g., the task name and the input and output data) to the task buffer in the corresponding task slot (step 2). As soon as writing the task buffer is finished, the same task slot ID is written to the APP_2_CM FIFO (step 3). The CoreManager reads this FIFO. It firstly performs a data-dependency checking among all tasks which are currently in the system. For this purpose, (1) must be evaluated for each transfer for all tasks. The array formed by pointer p_1 and size s_1 of task 1 is compared with the array formed by p_2 and s_2 of task 2. p_1, p_2, s_1 and s_2 are assumed to be greater or equal to zero. The equation is valid if a dependency is found.

$$dep = (unsigned)(p_1 - p_2) < s_2 \;\|\; (unsigned)(p_2 - p_1) < s_1 \quad (1)$$

In the particular case of the CM-EIS processor, the operations shown in (1) are merged into one instruction, which is thus executed in a single clock cycle. Furthermore, the application of 4-SIMD vectorization enables the execution of four parallel dependency checks. A more detailed explanation of the dynamic data-dependency checking of the CM-EIS processor can be found in [18].

If no data-dependency is found, the task is included in the *ready* task list. Otherwise, the task is annotated at the corresponding preceding tasks descriptions (step 4-6).

In the next step, the task-scheduling module selects the most suitable task from the *ready* task list (step 7). Two scheduling approaches are currently available. An *as-soon-as-possible* scheduling approach prioritizes the tasks according to their time of arrival in the CoreManager. The second possibility is an *earliest-deadline-first* approach, which favors tasks with the closest deadline. The scheduling is only performed if a suitable PE is available for the task.

After the scheduling process, a PE is allocated and local memory for the necessary data is reserved (steps 8-9). The implemented PE allocation approach is depicted in Fig. 5. The PE allocation is based on two bit masks: One corresponding to the PEs currently available and one corresponding to the PEs annotated as suitable for a task. The number of PEs determines the number of necessary bits (a dedicated bit is reserved for each PE). A value of one represents an available or suitable PE. An *AND* operation is performed on the bit masks representing the currently available and the suitable PEs. The PE associated to the first bit with a value of one (i.e., the first available and suitable PE) is subsequently allocated. In addition to this, for each task type the preferred and suitable PEs can be specified. The implemented PE allocation approach prioritizes preferred PEs accordingly. In order to increase data locality, a task can be scheduled on the same PE as its predecessor task, thus allowing the reuse of its output data. The number of memory transfers is hence reduced and the performance is improved.

A StartupUp Code is subsequently generated (step 10) by the CoreManager. It contains all necessary information to configure the PE (e.g., pointers to the instruction code) and all task data. It is transferred by two additional DMACs situated next to the CoreManager (step 11-12).

As soon as the task is finished, a packet is sent over the NoC and stored in the PE Finished FIFO (step 13). The CoreManager can evaluate this information (14-16). All successors of the executed tasks are put in the *ready* task list if no further dependencies are annotated (step 17). Finally, the corresponding task slot is made available for the APP by writing the task slot ID in the CM_2_APP FIFO (step 18).

 ISBN: 978-1-61208-308-7

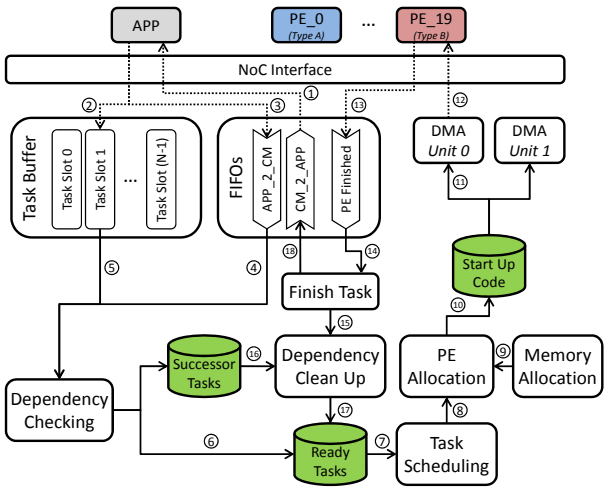


Figure 4. CoreManager structure and data flow

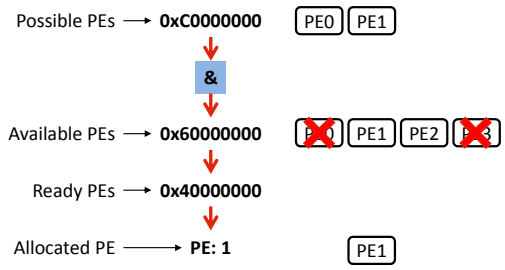


Figure 5. PE allocation

In the following, three versions of the CoreManager are compared and analyzed. The first one, called CM-LX4, is based on a Tensilica LX4 RISC core. The second solution integrates a Very Long Instruction Word approach in the CoreManager (CM-VLIW). The third version extends this processor with an improved instruction-set architecture especially suitable for the needs of a task scheduling unit (CM-EIS).

In Table I, the newly introduced instructions for the task scheduling are shown and shortly described. 16 task slots are always concurrently processed and can be hence evaluated in a single clock cycle. For each task slot, a validity bit is present. If it is set to a value of one the corresponding task slot is valid and can be used for the evaluation. The evaluation of the valid bit is included in the processing time of one clock cycle. In Fig. 6, the new instruction for finding the smallest values out of the *ready* task list is additionally shown. In this example, a minimum operator is applied. Nevertheless, it can be adapted at runtime to determine the maximum value. For each task slot, a 16-bit value must be defined. It can be flexibly used to specify, e. g., deadlines and priorities. The evaluation of all task slots is done in parallel.

TABLE I. TASK SCHEDULING INSTRUCTIONS

Instruction	Explanation
SCHED_SET(slot, val)	Value of a specified slot is set.
SCHED_SET_ALL(val)	All tasks slots are set to a specific value.
SCHED_MIN(slot, val)	Retrieves the smallest valid task slot. The task slot ID and its value are returned.
SCHED_MAX(slot, val)	As above, but the highest value is returned.
SCHED_INC(val)	Adds a value to all task slots. Task slot values saturates at 65535.
SCHED_DEC(val)	Subtracts a value from all task slots. Task slot values saturates at 0.

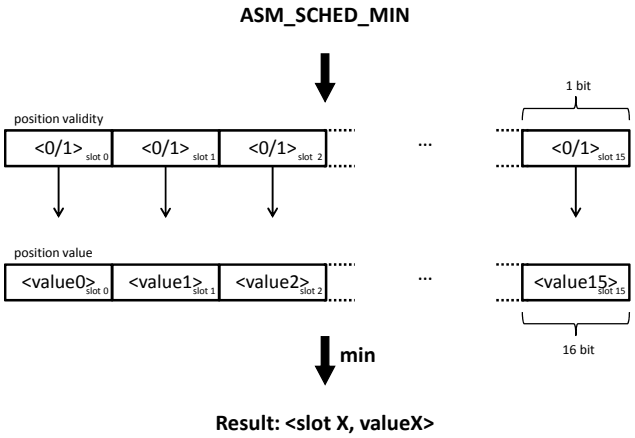


Figure 6. Instruction set architecture extension for task scheduling

V. ANALYTICAL MODEL

The developed analytical model depends on the following input parameters:

- n_{IN} : Number of input transfers
- n_{OUT} : Number of output transfers
- $n_{Transfers}$: $n_{IN} + n_{OUT}$
- n_{TIS} : Tasks currently available in the system
- n_{PE-ID} : Allocated PE number (starting with 0)
- $n_{Successor}$: Number of successor tasks

In the following, all components of the CoreManager will be analyzed and described using analytical methods. The timing is described in clock cycles. The CoreManager solely works on its local memory. Consequently, no external memory accesses are required and its processing time is hence independent of the clock frequency of the remaining system.

A. Dynamic Data-Dependency Checking

Equation (2) describes the necessary time for the dynamic data-dependency checking stage on the CM-LX4 and CM-VLIW processors. A quadratic dependence on the

 ISBN: 978-1-61208-308-7

number of output transfers is present. IN-IN transfer comparisons are not performed. In the case of independent tasks, no data dependencies have to be checked. Thus, $t_{Dep,\mathrm{var}}$ can be directly set to 0.

$$t_{DepCheck} = t_{Dep,init} + t_{Dep,TIS} * n_{TiS} + t_{Dep,Transfer} * n_{TiS} * \left(2 * n_{IN} + n_{OUT} \right) * n_{OUT} \quad (2)$$

In the case of the CM-EIS processor, the processing time of the dynamic data-dependency checking can be described by (3). IN and OUT transfers are not distinguished. Nevertheless, IN-IN transfers are not considered as a dependency. A quadratic dependence on the number of transfers is present.

$$t_{DepCheck,CM-EIS} = t_{Dep,init} + t_{Dep,TIS} * n_{TIS} + t_{Dep,Transfer} * n_{TIS} * \left(n_{Transfers} \right)^2 \quad (3)$$

B. Task Scheduling

The task scheduling finds the most suitable task from the *ready* task list. In the case of the CM-LX4 and CM-VLIW processor, the *ready* task list is sequentially searched for the smallest or largest value. For each task slot, the valid bit is evaluated. Equation (4) can be used to describe the processing time of this stage.

$$t_{Scheduling} = t_{Scheduling,const} + t_{Scheduling,\mathrm{var}} * n_{TIS} \quad (4)$$

In the case of the CM-EIS processor, the task scheduling time for up to 16 task slots is constant. Hence, (4) can be transformed to (5).

$$t_{Scheduling,CM-EIS} = t_{Scheduling,const} + t_{Scheduling,\mathrm{var}} * \left\lceil \frac{n_{TIS}}{16} \right\rceil \quad (5)$$

C. PE Allocation

Equation (6) determines the necessary time for the PE allocation.

$$t_{PE-Alloc} = t_{PE-Alloc,const} + t_{PE-Alloc,\mathrm{var}} * n_{PE-Id} \quad (6)$$

For the CM-EIS core up to 32 PEs can be evaluated in a single cycle. Hence, (6) can be modified as:

$$t_{PE-Alloc,CM-TIS} = t_{PE-Alloc,const} + t_{PE-Alloc,\mathrm{var}} * \left(1 + \left\lfloor \frac{n_{PE-Id}}{32} \right\rfloor \right) \quad (7)$$

D. Local Memory Allocation

Three different allocation approaches for the local memories are available. The *single-space* allocation occupies the whole memory for one task. The *top-down* allocation allows two tasks to use the same local memory. The most sophisticated mode of operation is the *block-based* allocation. The whole local memory is divided in equally sized blocks. In this case, eight blocks are used. The necessary processing time for the allocation of local memory is determined by (8).

$$t_{Mem-Alloc} = t_{Mem-Alloc,Init} + t_{Mem-Alloc,Transfer} * n_{Transfers} \quad (8)$$

E. DMAC configuration

The configuration time of the DMACs for transferring the Start Up Code is always the same. It can be described with (9).

$$t_{DMAC-Config} = t_{DMAC-Config,const} \quad (9)$$

F. Clean Up

The processing time after a task is finished depends on the number of successors per tasks. Additionally, the task slot ID must be written to the CM_2_APP FIFO. The processing time of the Clean Up stage can be expressed with (10).

$$t_{Clean-Up} = t_{Clean-Up,const} + t_{Clean-up,Successor} * n_{Successor} \quad (10)$$

G. System Level

A combination of the processing times of the components of the CoreManager leads to a system-level latency point of view. The processing time of the CoreManager for each part can be separated in a processing time before and after task execution. Equation (11) describes this behavior.

$$t_{Task-\mathrm{Proc}} = t_{Task-Start} + t_{Task-End} \quad (11)$$

Both terms on the right side of (11) can be individually expressed with (12) and (13), respectively. By using these equations it is possible to predict the performance of the CoreManager and determine its influence on the system.

$$t_{Task-Start} = t_{DepCheck} + t_{Scheduling} + t_{PE-Alloc} + t_{Mem-Alloc} + t_{DMAC-Config} \quad (12)$$

$$t_{Task-End} = t_{Clean-Up} \quad (13)$$

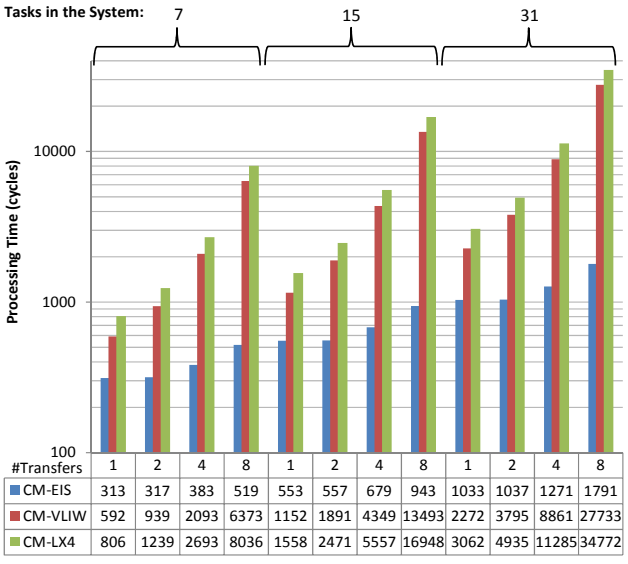
VI. RESULTS

In the first part of this section, the measured results of the CoreManager components are presented. Configurable task descriptions are used to measure the processing time. Especially corner cases are regarded. The FPGA prototype is used for all measurements. The integrated DebugUnit is responsible for generating traces at runtime. The DebugUnit is a dedicated component placed next to the CoreManager. It is used to observe the dynamic decisions of the CoreManager. The analysis of the traces is done with the DebugVisualizer. The processing time of the CoreManager components is deterministic due to the instruction and data fetch solely from its local memories. The same input leads to the same result and the same processing time. Due to this deterministic behavior, the presented results are valid for RTL and Netlist simulation as well as the ASIC prototype.

 ISBN: 978-1-61208-308-7

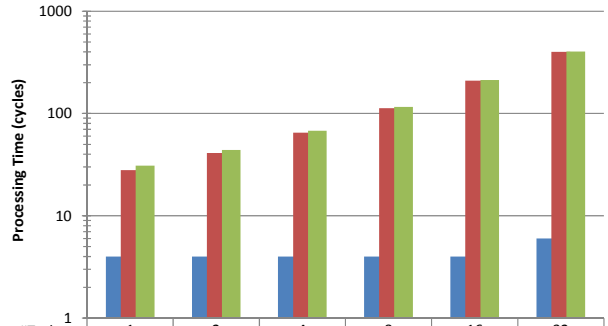
In the second part of this section, the previous results are analyzed to obtain the parameters of the analytical model. The last part of this section presents a comparison of the analytical model with the measurements of real applications.

In Fig. 7, the results of the dynamic data-dependency checking stage are depicted. All transfers are divided in 50 % input and 50 % output transfers. In the case of one transfer, an INOUT type is used. In the analytical model an INOUT transfer is regarded as an OUT transfer. n_{TIS} is varied between 7, 15, and 31. The number of transfers is set to 1, 2, 4, and 8. A difference in the processing time of over one order of magnitude can be observed between the CM-LX4 and the CM-EIS CoreManager. In Fig. 8, the processing time of the task scheduling is shown. The number of tasks in the *ready* task list is varied between 1 and 32. In Fig. 9, the results for the PE allocation are depicted. It is distinguished between the annotation of possible and possible/preferred PEs per task. The results for the local memory allocation are shown in Table II. The processing time depends on the already allocated blocks and on the number of transfers. The configuration of the DMA controller of the CoreManager needs a constant processing time of 12 cycles per task. In Table III, the processing time of the Clean Up stage is shown. For each successor task the necessary time is increased.



#Transfers	1	2	4	8	1	2	4	8	1	2	4	8
CM-EIS	313	317	383	519	553	557	679	943	1033	1037	1271	1791
CM-VLIW	592	939	2093	6373	1152	1891	4349	13493	2272	3795	8861	27733
CM-LX4	806	1239	2693	8036	1558	2471	5557	16948	3062	4935	11285	34772

Figure 7. Dynamic data-dependency checking results. The available tasks in the system and the number of transfers are varied.



#Tasks	1	2	4	8	16	32
CM-EIS	4	4	4	4	4	6
CM-VLIW	28	41	65	113	209	401
CM-LX4	31	44	68	116	212	404

Figure 8. Task Scheduling results. The number of tasks in the ready list is varied.

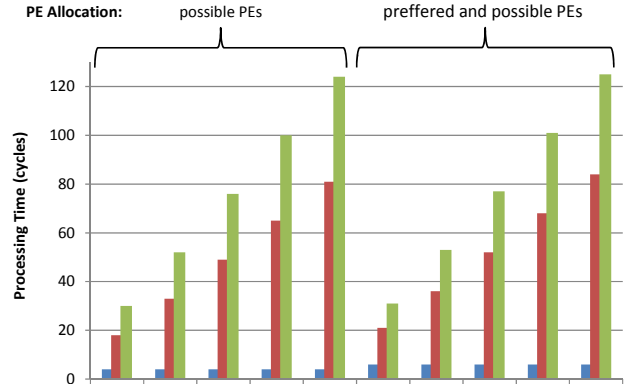
TABLE II. LOCAL MEMORY ALLOCATION: PROCESSING TIME (IN CYCLES)

Avail. Blocks	#Transfers	CM-EIS	CM-VLIW	CM-LX4
0x0	2	10	51	66
	4	20	56	78
	8	34	64	92
0x1	2	10	51	65
	4	20	56	77
	8	34	64	91
0x3	2	10	52	67
	4	20	57	79
	8	34	65	93
0x7	2	10	56	72
	4	20	61	84
	8	34	69	98
0x9	2	10	55	62
	4	20	55	74
	8	34	63	88
0x12	2	10	53	60
	4	20	53	72
	8	34	61	86

TABLE III. CLEAN UP: PROCESSING TIME (IN CYCLES)

#Successor Tasks	CM-EIS	CM-VLIW	CM-LX4
1	44	124	190
2	54	150	228
4	72	186	276
8	108	306	400

 ISBN: 978-1-61208-308-7



PE-ID	0	7	15	23	31	0	7	15	23	31
CM-EIS	4	4	4	4	4	6	6	6	6	6
CM-VLIW	18	33	49	65	81	21	36	52	68	84
CM-LX4	30	52	76	100	124	31	53	77	101	125

Figure 9. PE allocation results. PE-ID is varied.

In order to obtain the parameters described in section V the minimum mean square error is used. The resulting values for all parameters are presented in Table IV. The superior performance of the CM-EIS core, which was already observed in the first part of this section, is also noticed here. Especially $t_{Dep,Transfer}$, $t_{Scheduling,const}$, $t_{Scheduling,\mathrm{var}}$, $t_{PE-Alloc,const}$ and $t_{Mem-Alloc,Init}$ are significantly lower compared to those of the CM-LX4 and CM-VLIW cores. In the case of the local memory allocation, the parameter $t_{Mem-Alloc,Transfer}$ of the CM-VLIW core is smaller in comparison to the CM-EIS core due to constant processing time of the CM-EIS core. The data dependent processing time of the CM-VLIW core leads in average to a smaller value for parameter $t_{Mem-Alloc,Transfer}$. Nevertheless, the overall processing time of the CM-VLIW core is still 2 to 5 times higher (see Table II).

The corresponding relative errors are presented in Table V. The highest relative error corresponds to the dynamic data dependency checking stage. In the case of the CM-EIS core it is 6.3 %. These errors result from the data dependent execution time, .i.e., a dependency must be annotated at the predecessor of a task. Hence, an additional amount of processing time is needed. The DMAC configuration is for all cores perfectly predictable. Furthermore, the task scheduling and PE allocation models of the CM-EIS CoreManager have no error.

The relative errors for three real-world applications are depicted in Table VI. For each CoreManager approach the measured traces are compared with the prediction of the developed analytical models. The first two applications belong to the signal processing domain. In particular, the physical layer of a Global System for Mobile Communications (GSM) and Universal Mobile Telecommunications System (UMTS) are employed.

The third application is a JPEG decoding application. It decodes a picture with a resolution of 2560 by 1440 pixels. No data dependency checking is applied in the JPEG decoding application. Therefore, $t_{Dep,Transfer}$ is set to 0. Hence, no successor tasks are present in the Clean Up Stage. Each application is dynamically started several times.

All versions of the CoreManager have been synthesized with Synopsys Design Compiler for a 65 nm low power TSMC process using worst case conditions (125 °C, 1.08 V). For a target frequency of 333 MHz the occupied silicon area is 0.140 mm^2 (CM-LX4), 0.180 mm^2 (CM-VLIW) and 0.284 mm^2 (CM-EIS), respectively. Only logic area is evaluated, disregarding local memory area but including the memory interfaces (for timing correctness).

TABLE IV. MODEL PARAMETER

#Successor Tasks	CM-EIS	CM-VLIW	CM-LX4
$t_{Dep,init}$	118	107	110
$t_{Dep,TIS}$	32	68	92
$t_{Dep,Transfer}$	0.4	12.8	16.2
$t_{Scheduling,const}$	2.0	16.9	20
$t_{Scheduling,\mathrm{var}}$	2.0	12	12
$t_{PE-Alloc,const}$	2.0	22.0	32.0
$t_{PE-Alloc,\mathrm{var}}$	2.0	2.0	3.0
$t_{Mem-Alloc,Init}$	2	46.4	57.5
$t_{Mem-Alloc,Transfer}$	4	2.4	4.2
$t_{DMAC-Config,const}$	12.0	12.0	12.0
$t_{Clean-Up,const}$	34.9	99	161
$t_{Clean-up,Successor}$	9.2	25.5	29.7

TABLE V. MEAN RELATIVE ERROR COMPARED TO MEASURED VALUES FOR CONFIGURABLE TASKS FOR PARAMETER EXTRACTION

CoreManager Component	CM-EIS	CM-VLIW	CM-LX4
Data-Dependency Checking	6.3 %	3.6 %	2.5 %
Task Scheduling	0	0.8 %	0.9 %
PE Allocation	0	1.2 %	0.8 %
Local Memory Allocation	3.3 %	3.3 %	4.3 %
DMAC Configuration	0	0	0
Clean Up	0.6 %	2.4 %	1.3 %

TABLE VI. RELATIVE ERROR COMPARED TO MEASURED VALUES FOR REAL WORLD APPLICATIONS IN PERCENT (CM-EIS/CM-VLIW/CM-LX4)

CoreManager Component	Application		
	GSM	*UMTS*	*JPEG*
Data-Dependency Checking	3.6/3.2/3.9	6.9/4.3/2.6	0/0/0
Task Scheduling	0/0.2/0.4	0/0.3/0.6	0/0.3/0.3
PE Allocation	0/1.2/1.0	0/0.9/1.2	0/0.2/0.2
Local Memory Allocation	0/0/0	0/0/0	0/0/0
DMAC Configuration	0/0/0	0/0/0	0/0/0
Clean Up	0.4/0.2/0.4	0.3/0.9/0.4	0/0/0

 ISBN: 978-1-61208-308-7

VII. Conclusion and Future Work

In this paper, a central scheduling unit called CoreManager is analyzed. An analytical model has been derived from system analysis. A tool flow was introduced to generate the system and to obtain data traces. Parameters for all three CoreManager approaches have been derived from the analyzed data. It has been shown that the relative error on component level is less than 6.3 % compared to the measurements. On system-level with real application benchmarks, the relative error was shown to be lower than 6.9 %.

Future work aims at implementing a silicon prototype of the CoreManager in a heterogeneous MPSoC. Further optimizations of the architecture and the algorithms will be performed, especially regarding performance, area and power consumption.

Acknowledgment

The major part of this research work has been funded by the DFG through the cluster of excellence Center for Advancing Electronics Dresden and the European Union and the state of Saxony through the IMData project. A minor part was funded by the German Federal Ministry of Education and Research within the scope of the CoolBaseStations project.

Furthermore, we would like to thank Synopsys, Tensilica and Xilinx for sponsoring Software, IPs and prototyping FPGAs.

References

[1] K. Asanovic et al., “The landscape of parallel computing research: a view from Berkeley,” Technical Report UCB/EECS-2006-183, Electrical Engineering and Computer Sciences, University of California, Berkeley, Long Beach, CA, USA, Dec. 2006.

[2] E. A. Lee and D.G. Messerschmitt, “Synchronous data flow,” Proceedings of the IEEE, Vol.75, No.9, Sept. 1987, pp. 1235–1245, doi: 10.1109/PROC.1987.13876.

[3] C. R. Johns and D. A. Brokenshire, "Introduction to the Cell Broadband Engine Architecture," IBM Journal of Research and Development , Sept. 2007, vol.51, no.5, pp. 503-519.

[4] J. Glossner et al., "The sandbridge SB3011 SDR platform," Mobile Future, 2006 and the Symposium on Trends in Communications, SympoTIC '06, Joint IST Workshop, June 2006, pp. 2-5, doi: 10.1109/TIC.2006.1708006.

[5] T. Limberg et al., “A Fully Programmable 40 GOPS SDR Single Chip Baseband for LTE/WiMAX Terminals,” 34th European Solid-State Circuits Conference (ESSCIRC'08), Edinburgh, Great Britain, Sept. 2008, pp. 466-469, doi: 10.1109/ESSCIRC.2008.4681893.

[6] J. Lee, V. J. Mooney III, A. Daleby, K. Ingström, T. Klevin, and L. Lindh, “A comparison of the RTU hardware RTOS with a hardware/software RTOS,” ASP-DAC '03, Proceedings of the Asia and South Pacific Design Automation Conference, 2003, pp. 683-688, doi: 10.1109/ASPDAC.2003.1195108.

[7] P. Bellens, J.M. Perez, R.M. Badia, and J. Labarta, "CellSs: a Programming Model for the Cell BE Architecture," in SC’06, Proceedings of the Supercomputing conference, 2006, p. 86.

[8] M. Frigo, C. E. Leiserson, and K. H. Randall, “The implementation of the Cilk-5 multithreaded language,” Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation, June 1998, pp. 212- 223, doi: 10.1145/277652.27772.

[9] K. Fatahalian et al., “Sequoia: Programming the memory hierarchy,” IEEE Conference on Supercomputing, 2006, p. 4, doi: 10.1109/SC.2006.55.

[10] A. Ghuloum, E. A. E. Sprangle, and J. Fang, “Flexible parallel programming for Terascale Architectures with Ct,” Intel Technology Journal, vol. 11, no. 3, Aug. 2007, pp. 185-196.

[11] A. Wang, E. Killian, D. Maydan, and C. Rowen, "Hardware/ software instruction set configurability for system-on-chip processors," Design Automation Conference, 2001, pp. 184-188, doi: 10.1109/DAC.2001.156132.

[12] A. Chormoviti, N. Vassiliadis, G. Theodoridis, and S. Nikolaidis, "Enhancing Embedded Processors with Specific Instruction Set Extensions for Network Applications," Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2005, IDAACS 2005, Sept. 2005, pp.199-203, doi: 10.1109/IDAACS.2005.282969.

[13] J. Castrillon, D. Zhang, T. Kempf, B. Vanthournout, R. Leupers, and G. Ascheid, “Task Management in MPSoCs: An ASIP Approach,” International Conference on Computer-Aided Design, San Jose, California, USA, 2009, pp. 587-594.

[14] H. G. Rotithor, "Taxonomy of dynamic task scheduling schemes in distributed computing systems," Computers and Digital Techniques, IEE Proceedings, Jan 1994, vol.141, no.1, pp.1-10, doi: 10.1049/ip-cdt.19949630.

[15] O. Arnold and G. Fettweis, "Power Aware Heterogeneous MPSoC with Dynamic Task Scheduling and Increased Data Locality for Multiple Applications," Embedded Computer Systems (SAMOS), 2010 International Conference on, July 2010, pp. 110-117, doi: 10.1109/ICSAMOS.2010.5642075.

[16] M. Winter and G. Fettweis, “Guaranteed Service Virtual Channel Allocation in NoCs for Run-Time Task Scheduling,” Proceedings of the Design Automation and Test in Europe (DATE'11), Grenoble, France, March 2011, pp. 1-6, doi: 10.1109/DATE.2011.5763073.

[17] Tensilica Inc., www.tensilica.com, since March 2013 Cadence (http://www.cadence.com) [retrieved August 2013]

[18] O. Arnold, B. Nöthen, and G. Fettweis, “Instruction Set Architecture Extensions for a Dynamic Task Scheduling Unit,” Proceedings of the IEEE Annual Symposium on VLSI (ISVLSI'12), Aug. 2012, pp. 249-254, doi: 10.1109/ISVLSI.2012.51.

 ISBN: 978-1-61208-308-7

Practical Methodology for Adding New MANET Routing Protocols to OPNET Modeler

Rani Al-Maharmah, Guido Bruck, and Peter Jung
Department of Communication Technologies
University of Duisburg-Essen
Duisburg, Germany
e-mail: info@kommunikationstechnik.org

***Abstract*—Optimized Network Engineering Tool (OPNET) Modeler is a comprehensive development environment that enables users to model communication networks and distributed systems by performing discrete event simulations. The OPNET Modeler accelerates the design and model of Mobile Ad Hoc Network (MANET) routing protocols by providing tools for all phases, including model design, simulation, data collection, and data analysis. Using the already defined MANET routing protocols is straightforward, while modifying or extending them is a tricky process that can be time and effort consuming. This paper provides an overview of OPNET Modeler architecture and describes a practical methodology to add new MANET routing protocols to OPNET Modeler by using the Multi-Aware Cluster Head Maintenance (MACHM) as an implementation example.**

Keywords-communication systems modeling; simulation-based approach; OPNET Modeler; MANET routing protocols; MACHM

I. INTRODUCTION

The performance of communication systems can be evaluated using different approaches and techniques. Those techniques can be classified into: formula-based calculations, waveform-level simulations, and hardware prototyping and measurements [1]. It is obvious that the performance evaluation based on measurements obtained from hardware prototypes of designs is accurate and useful, especially in later stages of production. In general, this approach is very costly and time-consuming. Besides, it is not very flexible, especially in the earlier stage of the design when the number of design alternatives may be large. Instead, powerful computer-aided analysis and design tools can be used for the modeling and simulation of complex communication systems. Those computer-aided analysis and design tools can be classified further into: formula-based and simulation-based approaches.

Using the formula-based techniques, which are based on simplified models, offers a considerable insight into the relationship between the different design parameters and the resulted system performance. It is useful to use such techniques in the early stages of the design, because it enlarges the design space. On the other hand, it is extremely difficult to just use such techniques to evaluate the performance of complex communication systems with a high degree of accuracy. Simulation-based approaches enable any level of detail to be modeled with the chance of wider design space than the one that is possible with formula-based approaches or hardware prototypes measurements. In simulation-based approaches, it is possible to combine both mathematical and empirical models easily. A simulation-based approach can be used to produce designs that are: timely, cost-effective, and error-free. On the other hand, the major disadvantage of using the simulation approach is the resulted computational burden.

OPNET Modeler is a flexible and powerful commercial tool [2], which is used to analyze and design communication networks, devices, protocols, and applications. OPNET Modeler incorporates a broad suite of protocols and technologies. OPNET Modeler includes a development environment that used to enable modeling of different network types. This includes any network with mobile devices such as cellular, mobile ad hoc, wireless LAN, personal area networks, and satellite. OPNET Modeler uses a combination of state transition machine diagrams and C (or C++) codes to implement the different technologies. Those codes interact with the different state transition machine diagrams, which are defined in the different process models. The codes themselves are scattered in different physical and logical places. In OPNET Modeler the code can be found in header files, external files, process models, header blocks, function blocks, diagnostic blocks, termination blocks, and so on. The challenging task is to understand the structure of the simulator and being able to track and modify or add the different required construction parts. In case of adding new MANET routing protocols, it becomes even more challenging and frustrating, because of the amount and type of modifications needed to enable the network nodes to use this new modified protocol. With lack of such knowledge, it will be hard to take advantage of the powerful tools provided by the OPNET Modeler, especially for the model design, simulation, data collection, and data analysis parts. A practical methodology showing the way and steps to do such modifications is needed. It will ease the process of merging the new MANET routing protocols into the OPNET Modeler and allow researchers to take a full advantage of the software capabilities.

This paper provides researchers with a systematic and easy to accomplish way for adding new models to OPNET Modeler, such as new MANET routing protocols. In this paper, a detailed description of the steps required for

 ISBN: 978-1-61208-308-7

modeling and merging a new MANET routing protocol named Multi-Aware Cluster Head Maintenance (MACHM) into OPNET network simulation software is presented. This work is intended to help and guide other OPNET Modeler researchers, who are interested in studying and investigating new MANET routing protocols.

The rest of this paper is organized as follows. Section II briefs the new MACHM, while Section III provides an overview of the OPNET Modeler architecture and the MACHM simulation project cycle. Section IV describes in detail the methodology of adding the MACHM to the OPNET Modeler. Simulation study of MACHM is presented in Section V, while Section VI concludes the paper.

II. MULTI-AWARE CLUSTER HEAD MAINTENANCE (MACHM)

MANETs are self-organizing mobile wireless networks with a decentralized control of operations, which does not rely on a preexisting infrastructure like access point to communicate [3]. Network nodes have the ability of free movement around. Normally, each node has to act as a router in order to keep the network operating. MANETs can be used in different areas and applications, examples include military scenarios, rescue operations, conferences, any application that needs mobility, and areas where it is hard to build a wired network [4]. Typically, node's resources are limited and valuable in MANETs. Most importantly, the battery power because it is limited due to the relatively small size of mobile nodes. Managing this limited resource is a key challenge in MANET's environments.

Routing protocols can be classified to proactive (table driven) and reactive (on-demand) routing protocols [6]. Table-driven routing protocols try to maintain consistent, up-to-date routing information from each node to every other node in the network. One obvious problem is the resulted network overhead. This occurs because of the amount of information collected, especially when the number of network nodes is large. Additional disadvantage is the wasted resources used to collect unnecessary routing information, which neither used nor useful later because of the MANET's dynamic nature. The advantage of using table-driven technique is that the routes are known immediately when they are needed. On the other hand, on-demand routing protocols create routes only when they are needed. This results in a reduced routing overhead cost but at the expense of a route establishment delay.

In order to achieve scalability while maintaining a good routing, hierarchical or hybrid solutions are adopted, like the cluster-based concept. The main idea is to group the nearby nodes into logical groups known as clusters, then assigning nodes different functions inside and outside the group (cluster) [5]. Each group contains a special node, which acts as a leader of the group based on some criteria. Different cluster-based techniques use different basis to decide this special node, such as highest ID, lowest ID, node's connectivity, node's power level, or simply a random node. The special chosen node is used to label the cluster and to communicate to other nodes on behalf of the cluster [7]. Researchers refer to this special node with different terms. Some terms used to express the leader are: cluster head (CH) [8], coordinator [9], core [10], member of dominating set [11], and backbone network [12].

Another advantage of using the cluster-based concept (beside the scalability) is the ability to mix the two different techniques of routing. Proactive routing technique can be used inside the clusters, where the number of network nodes is relatively small. While reactive routing technique can be used outside, between the created clusters using the cluster head nodes as access points.

Electing the cluster heads and maintaining them throughout the progress of the ad hoc networks are vital and critical. The importance comes from the role they are playing during the network lifetime. A multi-aware approach, namely Multi-Aware Cluster Head Maintenance (MACHM), have been designed for electing the cluster heads and maintaining them. MACHM aims to reduce the total amount of the power consumed by the nodes in the network. Especially, the cluster head nodes that are more sensitive to power drain, because of their extra roles in the network. MACHM implements a cluster-based approach to achieve scalability and to take advantage of the hybrid routing technique.

MACHM involves in cluster head election, clusters formation stage, and cluster head re-election procedures. It invokes the cluster head election and clusters formation at the time of system activation. In MACHM, not all the nodes are allowed to participate in the cluster head election. Instead, the decision of participating or not will depend on the initial node's battery power value. The reason for allowing smaller set of nodes to participate is to ensure that the candidate's battery power will be reasonable and will not reach the re-election threshold quickly.

To be more efficient, many factors will be considered in all stages. MACHM takes into account the ideal number of nodes that a cluster can handle (load balancing consideration), the distance between the node and its neighbors (geographical consideration), the speed of nodes (mobility consideration), and most importantly the node's battery power (energy consideration). The cluster head election is based on a weighted formula that includes the previous factors as states in Formula (1).

$$W_n = \Delta_n \, w1 + L_n \, w2 + M_n \, w3 + P_n \, w4 \qquad (1)$$

where *Wn* is the weight value for node *n*, Δn is the degree difference for node *n*, *Ln* is the summation of the distances for node n with all its neighbors, *Mn* is the speed average for node *n*, *Pn* is the current battery energy consumption value for node *n*, and *w1*, *w2*, *w3* and *w4* are constant values used to decide the relative importance of each factor, where *w1+w2+w3+w4=1*. This weighted formula enables the usage of some or all factors in the cluster head election calculation, based on the network scenario or setting. By assigning zero to any constant from *w1* or *w2* or *w3* or *w4* it will ignore the related factor in the election calculation. The previous calculated weight values will be exchanged between nodes to determine the cluster heads.

 ISBN: 978-1-61208-308-7

The same method will be used later in the cluster head re-election procedure. A mechanism has been proposed to maintain the previously elected cluster heads based on their energy levels. If the elected cluster head battery power level reaches the defined threshold, it will then initialize a new cluster head election. The scope for the replacement cluster head will be from the set of direct neighbors. This is to reduce the effect of changing the cluster head and to use the already gathered information. Figure 1 shows the flowchart used for the weight calculation.

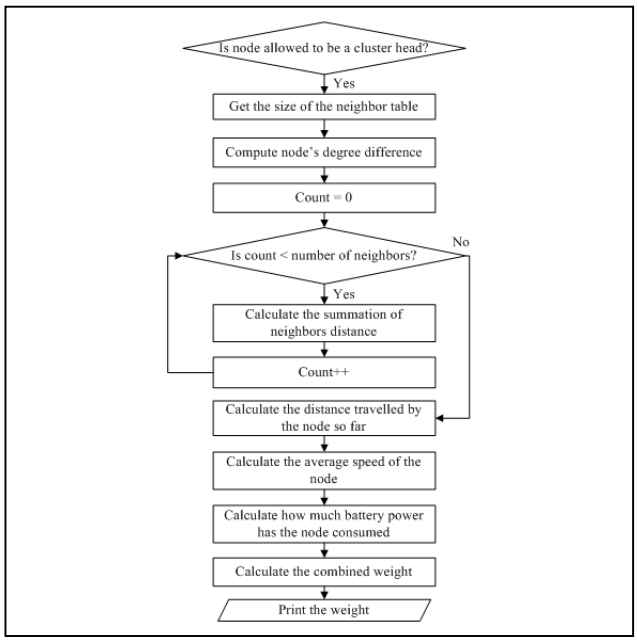


Figure 1. MACHM weight calculation flowchart

In Figure1, if the node's initial battery power level is over the defined threshold, then it is allowed to compete for the cluster head role. The node can immediately know the number of neighbor nodes by accessing the neighbor table and fetching its size. The node's degree difference is simply the abstract value of the number of neighbors subtracted from the *allowed_number_of_nodes* parameter, which is used for the load balance purpose. In MACHM, the nodes capsulate and send their coordinates in the *hello* messages. This enables the receiving nodes to calculate the distance and store this information along with the neighbor node. Each node then can calculate the distances summation with its neighbors, which gives an idea about the geographical position of each node.

In MACHM, each node tracks the distances it travels during the lifetime of the network. In a certain simulation time, it can calculate the speed it is travelling with by dividing the summation of distances by the current simulation time. This is the speed and mobility indicator used in MACHM. For the battery power consumed by each node, a battery energy consumption model is used. This model tracks and updates the battery power using the current draw values defined for the *SLEEP*, *IDLE*, *SEND*, and *RECEIVE* states. Finally, the node can calculate its weight value using Formula 1 and broadcast this value to its neighbors.

In order to implement the previously described method, network nodes need to gather different pieces of information relevant to their neighbors and keep them in different data structures. This collection of information can be done by exchanging control messages, which are broadcasted periodically or per event. Every control message intends to provide a certain piece of knowledge or invokes a certain action. MACHM uses route request, route reply, route reply acknowledgment, route error (link break detect, data packet no route, and route error received), hello, node weight, adjacent cluster head, and invoke request control messages. The complementary elements in MACHM are the timers and tables data structures. The timers used in MACHM are route entry invalid, route entry expired, route request expiry, connectivity loss, and cluster-head table timers. While the tables are route, IP common route, packet queue, route request, connectivity, adjacent cluster heads, and invoke request tables.

III. OPNET MODELER ARCHITECTURE

OPNET Modeler is a flexible and powerful tool, which provides a comprehensive development environment for the communication networks and distributed systems. It can be used to model and evaluate the performance of communication systems. OPNET Modeler contains a number of different construction tools. Each tool is concentrating on a specific phase or aspect of the modeling task. In OPNET Modeler, the MACHM simulation cycle is constructed from three major phases, namely: model specification, simulation and data collection, and analysis.

First step is to develop a representation of the intended system to be studied, this known as a model specification. OPNET Modeler environment provides different editors for the primitive building blocks. The available editors are project, node, process, external system, link model, packet format, ICI, and PDF editors. In OPNET Modeler, the model-specification editors are organized in a hierarchical fashion, which are network, node, process, and external system modeling environments.

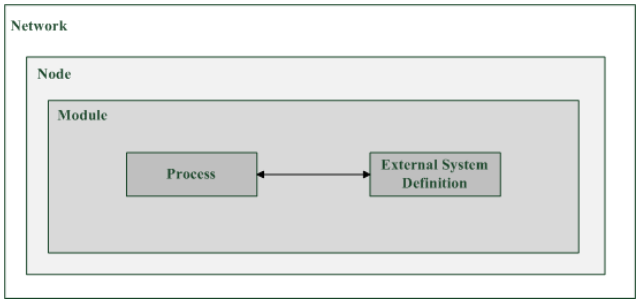


Figure 2. The relationship of hierarchical levels in models in OPNET

Figure 2 shows the relationship of the hierarchical levels in OPNET Modeler models. Network domain focuses on the network topology, which is described in terms of sub-networks, nodes, links, and geographical context. Node

 ISBN: 978-1-61208-308-7

domain focuses on the internal architecture of the node, which is described in terms of functional elements and data flow between them. Process domain focuses on the behavior of processes (protocols, algorithms, applications), which are represented in the form of finite state machines and extended high-level language (C or C++ code). External system domain focuses on the interfaces to models provided by other simulators running concurrently with a discrete event simulation.

Performance measurements of communication systems are the main goal of modeling them. This enables designers to study the behavior of such systems. Next step in MACHM simulation cycle is to run the simulations and collect the resulted data. OPNET Modeler allows a realistic estimation of performance and behavior for the executed simulations. It has several mechanisms to collect the desired data from one or more simulations of a system. OPNET Modeler uses discrete event simulations to produce different types of outputs (output vectors, output scalars, and animations). Users can define their own output file types as well, if this is desired. Normally, vast amount of output data will be generated after each simulation. Particular statistics or animations are explicitly activated in OPNET Modeler by recording them in the appropriate output files. This is done by specifying a list of probes when running a simulation. Each probe indicates that a particular statistic or form of animation should be collected in this run. Advance forms of probes can be defined by users in the probe editor.

The third and last phase of the MACHM simulation project cycle is analyzing and examining the collected results during the simulation. OPNET Modeler provides both a graphical and numerical processing environments, where user can investigate the generated results in depth. Additional data for plotting can be generated by a number of numerical processing operations in the analysis panels, including Probability Mass Function (PMF), Cumulative Distribution Function (CDF), histograms (occurrence and duration-based), confidence interval calculation, and mathematical filters defined in filter editor. In case of additional modifications for the communication system modeling, another round of the simulation cycle can be applied with the new specifications.

IV. IMPLEMENTING MACHM

In [13], a practical methodology for modeling wireless routing protocols using OPNET Modeler has been proposed. The authors implemented a modified wireless routing protocol named Geographical Ad-Hoc On-Demand Distance Vector (GeoAODV) as an implementation example. The methodology explains the way to modify an already existed routing protocol in OPNET Modeler. In case of adding a completely new routing protocol as MACHM, another practical methodology is needed, this is shown in this section.

In OPNET Modeler, the MANET is connected to the IP network through a MANET gateway that is running a MANET routing protocol and an IP routing protocol (or a static routing) on one of its interfaces. Figure 3 shows the node model of a MANET station in OPNET. MANET related files are gathered in the *manet* folder, except the header files, which are gathered in the *include* folder.

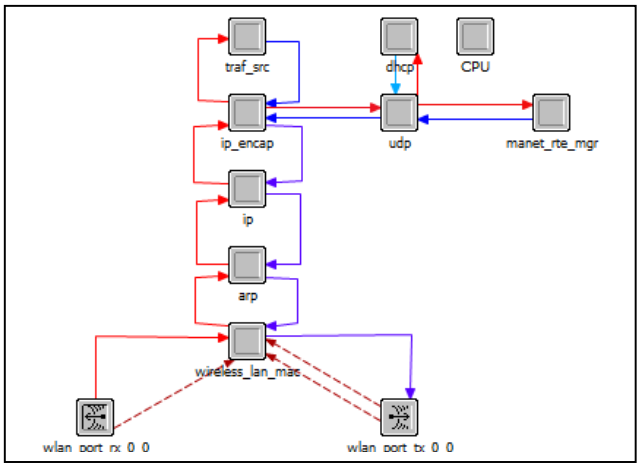


Figure 3. Node model of a MANET station in OPNET

The starting point to add a new MANET routing protocol to OPNET Modeler is the *manet_mgr.pr* process model. The MANET manager state machine functionality is to spawn the appropriate child routing process. This is based on the type of MANET routing protocol configured on the interfaces of the node. First, there is a need to register the newly MACHM routing protocol as a *manet_rte_protocol*. This is has to be accomplished in the header and function block of the *manet_mgr.pr* process model. Precisely, it is should be defined in the *manet_mgr_routing_protocol_determine* and *manet_mgr_routing_process_create* functions. A slight modification for *ip_higher_layer_proto_reg_sup.h* header file is needed as well to complete the registration process. A new child process named *machm_rte.pr* has been registered and added to the recognized list of the routing protocols in OPNET Modeler, as shown in Figure 4.

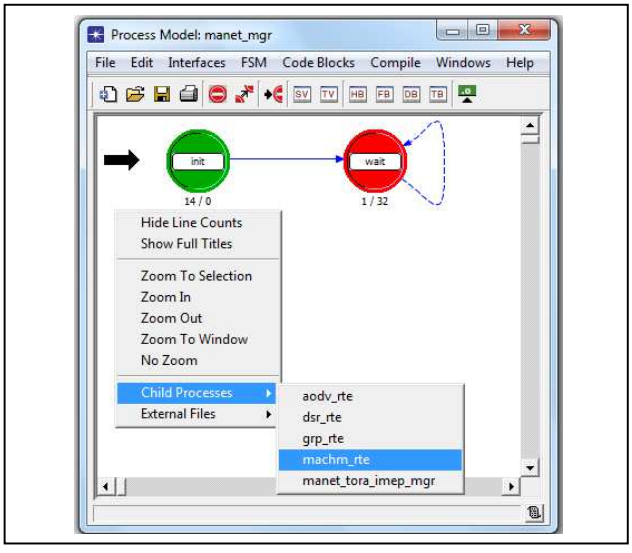


Figure 4. The modified *manet_mgr.pr* process model and its child processes

Next step is to define a new MACHM parameters group under the *AD-HOC Routing Parameters* group, which can be found in the *manet_mgr.pr* model attributes. The parameters are used to configure the behavior of the protocol. The parameters can be either a primitive or a compound type, it is also possible to control the range of parameter values and define a default ones. In order to make this new routing protocol available to the network devices, a change to the interface configuration of different types of nodes is needed. Like *manet_station_adv.nd*, *wlan_wkstn_adv.nd*, etc. nodes model.

The main work is done in the *machm.pr* process model, where the MACHM routing protocol functionality is defined by a finite state machine. The finite state machine initializes the required state variables and implements the different stages of the protocol. The implementation is achieved through the codes scattered in the entrance and exit parts, combined with the different blocks such as the header, function, termination, diagnostic blocks, and so on.

As stated before, the basic building blocks of MANET routing protocols are the control packets. MACHM packets format is defined using the packet format editor along with the external files. Normally, the definition contains functions to allocate and de-allocate the necessary memory needed in an explicit fashion. The memory allocation/de-allocation functions depend on the data types and storage sizes defined for the different control packets fields. The other important data structure in MANET routing protocols is tables. Tables are used to store useful information about the network, like the connectivity, data routes, data packets, and so on. The definition and handling of such data structures can be accomplished using the external files. A good practice is to define a single external file for each data structure (table) to ease the modification process in later stages.

For the communication systems performance evaluation purpose, different kinds of statistics are needed. OPNET Modeler offers two levels of statistics, local and global statistics. The local statistics can be collected on every node in the network, while the global statistics can be collected for the whole network. After defining the needed statistics in the *machm_rte.pr* process model, a registration of them is needed. The registration takes place in the function block using the *op_stat_reg* command, while the *op_stat_write* command is used to record the values during the simulation.

As mentioned before, in OPNET Modeler the structure of MANET routing protocol is scattered through different parts and files. First, MACHM defines the state and temporary variables, which are needed inside the routing protocol. They are used to hold some MACHM parameters or simply they provide a global scope for the different procedures and functions. The header and function blocks are responsible for the complete actions taking place in the MACHM routing protocol. Actions include: state variables initialization, battery energy level initialization, statistics registration, packets sending and arrival handling, updating the different tables, electing the cluster heads and so on. The defined finite state machine represents the connecting point for all those pieces. It makes the required transitions and calls of the different procedures and functions.

Instead of configuring the running routing protocol on node's interface one by one, another practical modification can be done to the *wireless_deploy_wiz_helper.xml* file. A couple of HTML code lines will include the newly defined MACHM routing protocol in the wireless deployment wizard routing protocols drop list, which then can be used directly in the project editor.

The practical methodology for adding a new MANET routing protocol to OPNET Modeler can be summarized as follows:

- Register the new MANET routing protocol through the *ip_higher_layer_proto_reg_sup.h* header file.
- Modify the *manet_mgr_routing_protocol_determine* and *manet_mgr_routing_process_create* functions in the *manet_mgr.pr* process model. This is to enable the network nodes to configure the new MANET routing protocol on their interfaces.
- Add the new MANET routing protocol as a child process to the *manet_mgr.pr* process model.
- Define the new MANET routing protocol parameters as a new group under the *AD-HOC Routing Parameters* group in the *manet_mgr.pr* model attributes. Additional levels of nesting for the parameters can be done too.
- Define the local and global statistics for the new MANET routing protocol through the header files.
- Register the newly defined statistics in the OPNET Modeler. The global statistics registration can be done in the header file, while the local ones in the function block of the process model.
- Configure the interface of the MANET and WLAN nodes by adding the new defined statistics.
- Define the finite state machine with the required transitions and codes.
- Define the state and temporary variables that will be used through the different blocks.
- Define the control packets and data structures needed for the new MANET routing protocol through the packet format editor and external files.
- Program the new MANET routing protocol through the header, function, diagnostic, and termination blocks.
- Modify the *wireless_deploy_wiz_helper.xml* file to enable the usage of the new MANET routing protocol by the wireless deployment wizard.

V. SIMULATION STUDY OF MACHM

After modeling MACHM in OPNET Modeler, a data collecting and analyzing is needed to complete the MACHM project cycle. In this section, a simulation and performance study of MACHM is presented. One important point is to decide which performance metrics better clarify the behavior of the new MANET routing protocol. In MACHM, the following performance metrics are considered: the number of weight and invoke request messages sent, the number of cluster heads, the average time a cluster head survives, and the energy consumption of the network nodes.

 ISBN: 978-1-61208-308-7

Network overhead can be measured in term of control messages sent in the network, such as the *MACHM_WEIGHT* and *MACHM_INVOKE_REQUEST* control messages. MACHM uses the weight messages to elect the cluster head, which is chosen from the set of the allowed nodes to compete. If the initial battery level is over the threshold, the node is allowed to participate in this competition. While the concept of invoke requesting is based on the battery power factor. An elected cluster head informs the nearby nodes that a re-election of the cluster head is needed, when its battery power level goes below the defined threshold. Another performance indicator is the number of cluster heads that exist in the network during its lifetime. It is important to have an idea about the average time an elected cluster head survives before it asks for a replacement. Saving the battery power of all nodes is the main concern of MACHM. Specially, the cluster heads that should not be loaded too much to ensure longer network life time. All nodes energy consumption metric is used to study this property.

Several combinations of OPNET Modeler and MACHM setups have been tested. Table I and Table II show the selected settings for this section. The changing parameter is the battery power level threshold, while the four factors have an equal importance of 25% in this scenario.

TABLE I. OPNET SIMULATIONS SETUP

Simulation Setup	Value
Number of Nodes	10
Area	100 X 100 m campus
Distribution of Nodes	Random
Mobility Model	Random Waypoint (1 m/s)
Operational Mode	802.11b
Data Rate	11 Mbps
Data Traffic	Complete VoIP mesh between nodes
Simulation Time	3600 seconds

TABLE II. MACHM PARAMETERS SETUP

MACHM Parameter	Value
Allowed Number of Nodes	3
Degree Difference Weight	0.25
Distance Summation Weight	0.25
Mobility Weight	0.25
Battery Consumption Weight	0.25
Battery Power Threshold	20%, 30%, 40%, 50%, 60% and 70%
Threshold Decrement	5%

As mentioned before, MACHM does not allow all network nodes to calculate and send their weight values. Only the nodes with initial battery power levels larger than the threshold are allowed to compete for the cluster head election. The tested threshold values are 20%, 30%, 40%, 50%, 60%, and 70% of the node's battery power level. Changing the threshold value will affect the number of weight messages send by the candidate nodes, as shown in Figure 5. In the simulations, the starting battery power level for each node is chosen randomly between 1% and 100% for more realistic scenarios. Increasing the threshold value will decrease the chance of having nodes with a starting battery power level over the threshold, and hence the weight messages send. When the threshold value is somehow high, for example 70%, only two nodes satisfy the condition, which means that only two weight messages are sent. As can be seen from Figure 5, reducing the threshold value allows more nodes to participate. The allowed nodes are five when the threshold is 60%, six for 50%, seven for 40%, eight for 30%, and nine for 20%.

According to the MACHM functionality, the winning cluster head continues its role in a normal way until its battery power level reaches the threshold. When this occurs, a cluster head reduces the threshold value by a predefined *battery_power_threshold_decrement* value. Also, it sends an invoke request message to all its neighbors. Receiving an invoke request message changes the behavior of MACHM. Now all the nodes are allowed to participate in the cluster head election process. In this simulation scenario, the heavy data traffic and the relatively small initial battery power levels for some nodes cause them to shut down earlier. In Figure 5, the second point of each line shows the number of network nodes that still alive in that simulation time.

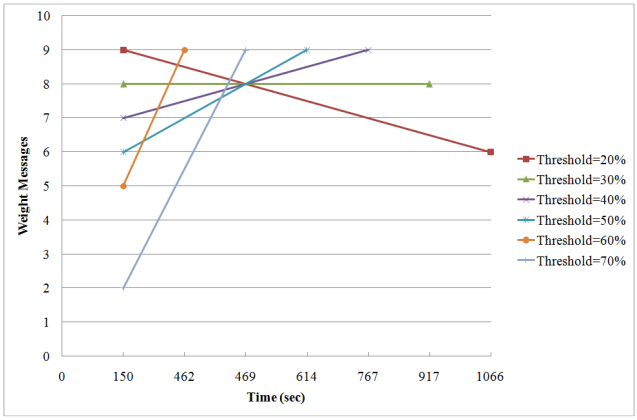


Figure 5. Weight messages sent during the simulation

On the other hand, the invoke request messages are used to notify the neighbor nodes that the current cluster head is looking for a better replacement (if available). This is why all the nodes will participate in the next cluster head election. Figure 6 shows that only one invoke request message is sent, which means that the new elected cluster head could survive till the end of the simulation. The chosen threshold affects how long the initial cluster head can survive in the network

 ISBN: 978-1-61208-308-7

before asking for a replacement. In general, it is noticeable that the lower the threshold value is the longer a cluster head survives and the later an invoke request message is sent. Figure 6 shows that the mechanism used by MACHM to elect the cluster heads is efficient, since it avoids multiple sending of invoke request message for the same threshold.

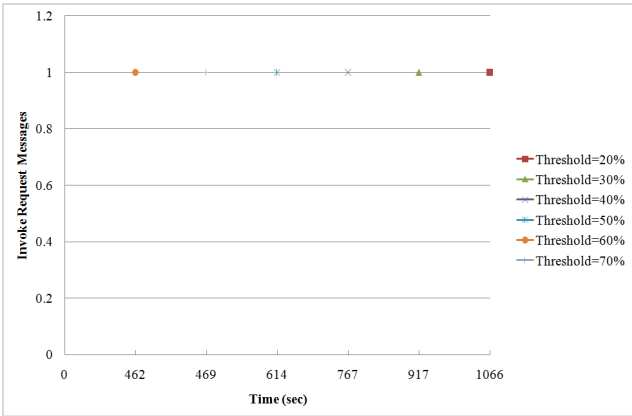


Figure 6. Invoke request messages sent during the simulation

In networks with large number of nodes, the number of cluster heads exist in the network gives a good idea about the balance and the distribution of the nodes. In this section, the chosen setup produces a connected network with no gaps. Figure 7 shows that only one cluster head exists for all the thresholds. The different points show the simulation time a new cluster head is introduced into the network. The first cluster head is chosen after some initializations, calculations, and weight messages exchange. The other cluster head is introduced into the network after receiving an invoke request message from the previously winning cluster head.

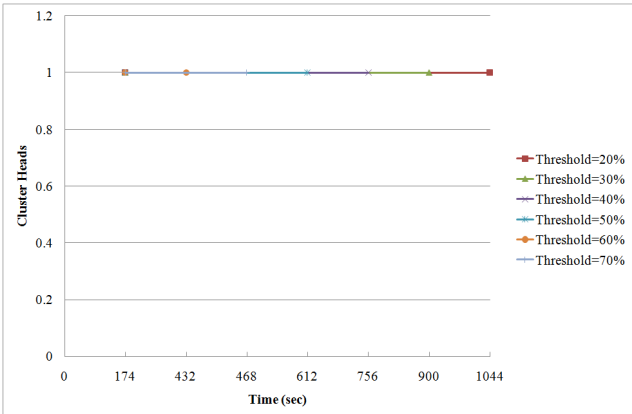


Figure 7. Total number of elected cluster heads during the simulation

MACHM aims to elect a cluster head that can survive for longer time to avoid the need of changing it frequently. This is can be achieved by implementing the multi-aware concept. MACHM makes a good combination of load balance, geographic distribution, speed, and power factors. Figure 8 shows the survive time for the first elected cluster head. It is noticeable that the survive time decreases when the threshold value increases. This is because the cluster head takes less time to reach the defined threshold, and then issues a re-election request to the neighbor nodes.

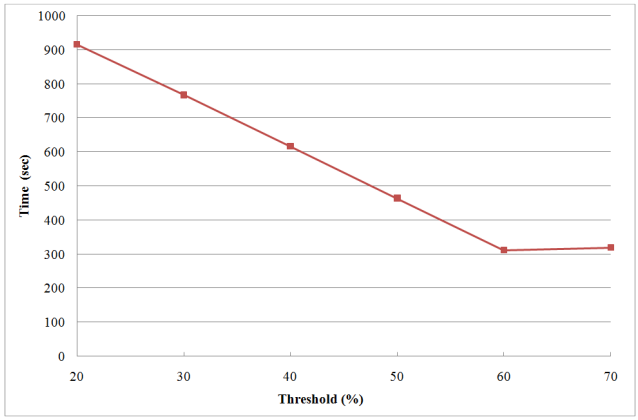


Figure 8. The survive time for the first elected cluster head

In OPNET, the information about node's battery power is not available as the other parameters. For this reason, MACHM creates its own battery energy consumption model. The created energy consumption model initializes the node's battery power levels with values between 1% and 100%. Also, it tracks and updates them through the life time of the network. The tracking and updating of those values is controlled by the current draw values. MACHM defines current draw values for the different node states, which are *SLEEP*, *IDLE*, *SEND*, and *RECEIVE* states.

Figure 9 shows the all nodes energy consumption for the VoIP simulation. The energy consumption varies according to the call buckets, showing the relatively low consumption values for *SLEEP* and *IDLE* states. While higher consumption values are recorded for the *SEND* and *RECEIVE* states.

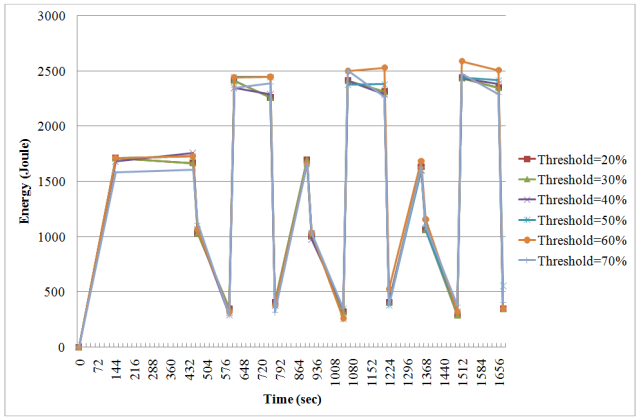


Figure 9. All nodes energy consumption during the simulation

This section showed the simulation results obtained for the new MACHM using the OPNET Modeler. The protocol itself is flexible and adjustable. It is important to select a

 ISBN: 978-1-61208-308-7

suitable combination of MACHM parameters setup to take advantage of its capabilities.

VI. CONCLUSION

OPNET Modeler is an important tool, used widely to model and study communication systems. MANETs attract many researchers because of the wide applications they can be used for. The aim of this paper is to guide the researchers who are interested in modeling and studying new MANET routing protocols using the OPNET Modeler. This paper proposed a practical methodology for adding such new MANET routing protocols to OPNET Modeler. It uses MACHM as an implementation example. A brief description of MACHM and the multi-aware concept is given. Also, the OPNET Modeler architecture is introduced for better understanding. The paper gives a detailed implementation of MACHM according to the proposed practical methodology, followed by general guidelines. The simulation study shows the performance evaluation of the implemented MACHM. Applying the ideas in this paper will accelerate the developing of new MANET routing protocols. This is because the researchers can take advantage of the powerful tools provided by the OPNET Modeler. This includes model design, simulation, data collection, and data analysis phases.

REFERENCES

[1] Michel C. Jeruchim, Philip Balaban, and K. Sam Shanmugan, “Simulation of Communication Systems: Methodology, Modeling, and Techniques,” New York: Kluwer Academic Publishers, October 2002, ISBN-10: 0306462672.

[2] http://www.opnet.com [retrieved: August, 2013].

[3] Yoav Sasson, David Cavin, and Andre Schiper, “A Location Service Mechanism for Position-Based Multicasting in Wireless Mobile Ad hoc Networks,” In Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 9, vol. 9, 2005, pp. 321b-321b.

[4] L. A. Latiff, A. Ali, Chia-Ching Ooi, and N. Fisal, “Location-Based Geocasting and Forwarding (LGF) Routing Protocol in Mobile Ad Hoc Network,” In Proceedings of the Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunications Workshop (AICT-SAPIR-ELETE '05), IEEE Computer Society, Washington, DC, USA, 2005, pp. 536-541.

[5] X. Hong, K. Xu, and M. Gerla, “Scalable routing protocols for mobile ad hoc networks,” IEEE Network, vol. 16, no. 4, 2002, pp. 11–21.

[6] Petteri Kuosmanen, “Classification of ad hoc routing protocols,” Finnish Defence Forces, Naval Academy, Finland, 2002. Available from http://www.netlab.tkk.fi/opetus/s38030/k02/Papers/12-Petteri.pdf.

[7] P. Sinha, R. Sivakumar, and B. Vaduvur, “Enhancing Ad Hoc Routing with Dynamic Virtual Infrastructures,” In Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, 2001, pp. 1763-1772, Anchorage, USA.

[8] Z.J. Haas, and S. Tabrizi, “On some challenges and design choices in ad-hoc communications," Military Communications Conference, MILCOM 98. Proceedings., IEEE , vol. 1, October 1998, pp. 187-192.

[9] Benjie Chen, Kyle Jamieson, Hari Balakrishnan, and Robert Morris, “Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks,” Wireless Network Journal, vol. 8, issue 5, September 2002, pp. 481-494.

[10] R. Sivakumar, P. Sinha, and V. Bharghavan, “CEDAR: a core-extraction distributed ad hoc routing algorithm,” Selected Areas in Communications, IEEE Journal on , vol. 17, no. 8, August 1999, pp. 1454-1465.

[11] Jie Wu, Ming Gao, and I. Stojmenovic, “On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks,” IEEE/KICS Journal of Communication Networks, vol. 4, no. 1, 2002, pp. 59-70.

[12] B. Liang, and Z.J. Haas, “Virtual backbone generation and maintenance in ad hoc network mobility management,” INFOCOM 19th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE , vol.3, March 2000, pp. 1293-1302.

[13] V. Hnatyshin, H. Asenov, and J. Robinson, “Practical methodology for modeling wireless routing protocols using OPNET Modeler,” In Proceedings of 21st International Conference on Modeling and Simulation (MS 2010), July 15 - 17, 2010, Banff, Alberta, Canada.

 ISBN: 978-1-61208-308-7

Combining Genetic Algorithms and Simulation to Search for Failure Scenarios in System Models

Kevin Mills, Christopher Dabrowski, James Filliben and Sandy Ressler
{kmills, cdabrowski, jfilliben, sressler}@nist.gov
Information Technology Laboratory
NIST
Gaithersburg, MD, USA

***Abstract*—Large infrastructures, such as clouds, can exhibit substantial outages, sometimes caused by failure scenarios not predicted during system design. We define a method for model-based prediction of system quality characteristics. The method uses a genetic algorithm to search system simulations for parameter combinations that result in system failures, so that designers can take mitigation steps before deployment. We apply the method to study an existing infrastructure-as-a-service cloud simulator. We characterize the dynamics, quality, effectiveness and cost of genetic search, when applied to seek a known failure scenario. Further, we iterate the search to reveal previously unknown failure scenarios. We find that, when schedule permits and failure costs are high, combining genetic search with simulation proves useful for exploring and improving system designs.**

Keywords—genetic algorithms; model-based prediction; simulation methodology; system design

I. INTRODUCTION

Modern society grows increasingly dependent on large infrastructures, such as clouds, for computation and storage needs. Yet, such infrastructures are prone to substantial outages, which sometimes arise from failure scenarios that were not considered during system design [1]. Because the state-space of such systems is vast [2], discovering potential failure scenarios is quite difficult, somewhat like searching for a needle in a haystack. A collection of evolutionary computation methods [3] exists to search for optimal solutions in large spaces. Among those methods, genetic algorithms (GAs) [4] provide a flexible approach, well suited for problems where little information is available about the structure of the solution space. Here, we investigate combining a GA with simulation to search system designs for *anti-optimal* solutions, e.g., parameter combinations that yield degraded performance, such that only a small percentage of a system's users can successfully obtain needed resources. We demonstrate that this approach can help system architects to identify potential failure scenarios before system deployment.

Section II presents related work. Section III illustrates our approach; describes the GA used in our case study, including key control parameters; discusses minimum requirements for a system simulator to be controlled by a GA; and outlines an iterative process to hunt for failure scenarios. Section IV describes a case study, applying the GA to search for a known failure scenario in an existing cloud simulator. In addition to introducing the simulator and related parameters, we explain how the GA maps simulator parameters to 'chromosomes'. Further, we illustrate our deployment of GA search as a distributed application on a cluster, and we define settings used for key GA control parameters. Section V presents and discusses results from our case study, illustrating the dynamics, quality, effectiveness and costs of GA search for a known failure scenario. We also discuss previously unknown failure scenarios discovered in subsequent repetitions of GA search. Section VI gives conclusions and future work.

II. RELATED WORK

Modern information systems are increasing in complexity: growing in size and geographic scope, changing constantly as software and hardware components are added and removed, and providing shared support to users with many different applications. These traits make failure scenarios both difficult to foresee and expensive to experience; thus, researchers actively pursue prediction techniques that can be applied during system design (offline) and at run time (online).

Available research literature generally explores design-time methods for complementary purposes: (1) improving system models or, like the current paper, (2) exploring system models to identify potential failure scenarios. One main goal of improving system models is to provide better probability estimates for rare events. Better estimates can help to parameterize system models with more realistic failure distributions. Researchers investigate two main approaches: splitting [5-7] and importance sampling [8], or some combination [9]. The main goal of such techniques is variance reduction for probability estimates of rarely occurring events. The main mechanism is to steer simulations into regions that generate more samples of rare events, which would otherwise occur infrequently and, thus require long simulation times in order to generate accurate estimates.

Some researchers use system models to search for failure scenarios that might otherwise be overlooked during design. For example, Shultz and colleagues [10] used an approach similar to ours, applying a GA to seek combinations of faults that cause anomalies in control behavior within two simulators, an autonomous aircraft and a submersible. Their work differs from ours in two main ways. First, they devised domain-specific encodings to match fault-scenario languages that were unique to each simulator. Our method uses classical GA binary encoding that can be applied generally to any simulator that can be parameterized numerically. Second, they used domain-specific knowledge to modify the usual genetic operators. We

 ISBN: 978-1-61208-308-7

applied a classical GA without modifying the crossover and mutation operators. Yucesan and Jacobson [11] used simulated annealing (SA) to search for event sequences leading to failed termination conditions. They report that SA has a major drawback: requiring customization of control parameters to suit particular problems. Our approach used GA control parameters selected [12] independently of specific problems. Dabrowski and Hunt [13] used graph analysis to search for cut sets (indicating failure vulnerabilities) in Markov chain models, derived from detailed system simulators. After identifying vulnerabilities, they used perturbation analysis to determine failure thresholds and trajectories. Their method involved modeling processes as zero-order Markov chains, which are "memoryless", and thus do not capture behavioral history. Our approach directly explores detailed system simulators, including historical behaviors, allowing assessment of detailed system processes. Fainekos and colleagues [14] propose a variety of optimization techniques (e.g., GAs, ant-colony optimization, Monte Carlo simulations and cross-entropy method) to search for parameter combinations that violate invariant execution trace properties expressed using Metric Temporal Logic. Exploiting such approaches requires a specification of desired behaviors in order to look for violations. Our approach requires only a single measure of anti-fitness, which does not require a rigorous statement of desired system properties.

While design-time methods seek failure scenarios that could arise after deployment, run-time methods aim to make specific, timely predictions of impending failures in deployed systems, so that system operators can take remedial actions. As Matsuo notes [15], predicting future behavior in complex systems appears quite difficult. Yet, even moderate success can have large positive returns, which encourages researchers to continue investigating the efficacy of many approaches [16] that might predict failures during system operation. Most researchers apply offline techniques, such as machine learning [17-18], hidden Markov models [19], GAs [20], and Bayesian estimation [21-22] to learn patterns, which online systems can monitor to predict failures. Other researchers [23-24] explore monitoring techniques without an offline component.

III. METHOD

Figure 1 gives a schematic of our method, where a GA controls a population of simulators distributed on a high-performance computing cluster. First, an analyst selects a list of simulator parameters to search, defining for each a range and granularity; thus each parameter can take on a discrete set of values. The GA uses this information to construct an internal representation, or 'chromosome' map, specifying the location and number of bits representing each parameter (see Table II, for an example). Subsequently, the GA instantiates random bit strings (i.e., chromosomes) for each simulator in the population, and then transforms them to parameter files. Each simulator reads its assigned file, updates its internal parameters accordingly and runs a simulation. As each run finishes, the corresponding simulator reports a metric, which we call *anti-fitness*, defined by an analyst to represent a measure of system failure. For example, in our case study (Section IV), we define anti-fitness as the proportion of users who could not be served by a simulated cloud. After collecting anti-fitness reports from the current population, the GA uses an algorithm [12] to construct a next generation of parameter combinations, and distributes a combination to each simulator in the population. Over multiple generations, tuples are collected (for later analysis) giving parameter settings and corresponding anti-fitness from each simulation run.

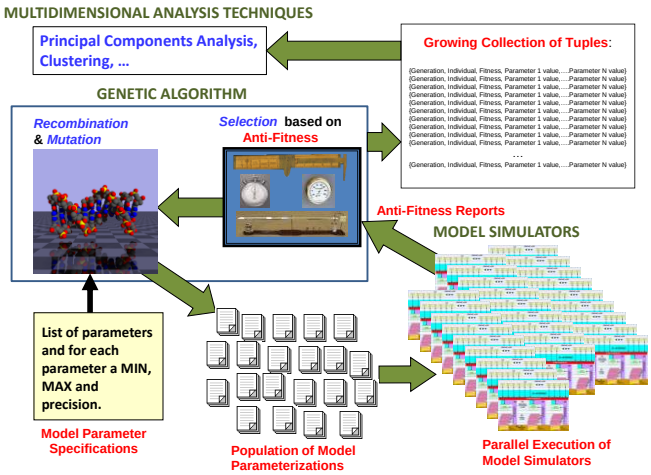


Figure 1. Schematic of a GA steering a population of simulators.

As generations progress, the GA steers the population of simulators toward parameter combinations that maximize the defined anti-fitness metric. GA steering is based solely on anti-fitness measures achieved by each set of bit strings (i.e., chromosomes). The search process is blind to the existence of parameters. The GA searches only for bit strings that achieve maximum anti-fitness. The connection between bit strings and model parameters occurs when the GA converts bit strings to parameters for input to simulators. For that reason, a GA is quite general and flexible, as we explain next.

The GA begins by generating randomly a seed population of individuals, each consisting of an appropriate length bit string, representing values for the chosen parameters of a problem to be solved. The *population size* is a control parameter of the GA. The GA next evaluates the anti-fitness of each individual. Over time, the GA evaluates the anti-fitness of many populations, where each population is called a generation. The population for generation n+1 is created through some transformation of individuals composing generation n. The GA terminates after a specified number of generations, unless terminated manually.

After evaluating a generation, the GA determines whether the population should be rebooted, which involves randomly regenerating all or part of the next generation. Rebooting can increase the GA's exploration of the search space. The GA includes a control parameter, *reboot proportion*, which defines the percentage of generations to complete before a reboot.

When selecting individuals for generation n+1, those with highest anti-fitness (i.e., the elite) from generation n can be included unchanged. The GA has a control parameter, *elite selection percentage*, which defines how many individuals from generation n will be placed unaltered into generation n+1. Such elite individuals can be placed into a population whether

 ISBN: 978-1-61208-308-7

or not the remaining individuals will be generated randomly or by transformation. When the population for generation *n*+1 is created by transformation, the procedure involves selecting a pool of candidate individuals from generation *n*, and then applying two genetic operators, crossover and mutation, which mimic reproduction in biological populations.

The GA includes a control parameter, *selection method*, which defines the algorithm used to select individuals from generation *n* for inclusion into the candidate pool, where the more anti-fit individuals from generation *n* may be included multiple times, while some of the least anti-fit individuals may be excluded. Given a pair of individuals, chosen randomly from the candidate pool, a GA control parameter, *number of crossover points*, determines where bits will be swapped between them. The specified number of crossover locations is chosen randomly, uniformly distributed within the length of bit strings comprising chromosomes, and the bits in each individual are swapped at those points and the two transformed individuals are placed into the population of recombined individuals. This continues until a sufficient number of (possibly) transformed individuals are selected to fill a population.

Subsequently, the GA iterates over each bit representing each recombined individual, while deciding whether or not the bits should be inverted. A GA control parameter, *mutation rate*, specifies the probability that any given bit will be flipped. After mutation, the GA inserts the individual into the population for generation *n*+1. Subsequently, the anti-fitness of each individual is evaluated and the GA iterates through generation *n*+1, creating the population of individuals for generation *n*+2, and so on until termination.

For a GA control, a simulation model must be able to initialize its parameters from external inputs and must be capable of executing in a loop, as simulations finish and new inputs arrive. Most simulators have such capabilities, but must be modified to asynchronously await inputs generated by a GA and to report anti-fitness once the simulation completes. We made these modifications via a signal file shared between each simulator instance and the GA. As discussed in Section V.A, more substantive simulator modifications were also necessary.

Once a complete search is finished, several potential failure scenarios may be revealed (see Section V). The analyst is then free to resolve those failures and repeat the GA search for additional scenarios. This iterative process can continue until no more failure scenarios appear or available time is exhausted.

IV. CLOUD SYSTEM CASE STUDY

We evaluated GA search while seeking a previously known failure scenario [25] in Koala, an existing infrastructure-as-a-service (IaaS) cloud simulator. The simulator architecture is shown in Figure 2. A full description of Koala can be found elsewhere [26-27]. Here, we give only a summary.

Koala simulates five layers: (1) demand from users, each requesting a collection of virtual machines (VMs), (2) a supply of physical nodes on which VMs can be placed, (3) a resource allocation layer, consisting of a cloud controller and cluster controllers that cooperate to determine a cluster on which to place VM collections, (4) an Internet/Intranet layer providing communication among simulated nodes, and (5) a VM behavior layer that models variations in resource usage over time. Available VM types and physical platforms are modeled after the Amazon EC2 Cloud, while the three-tier cloud architecture (cloud, cluster and node controllers) is modeled after a public domain version (1.6) of Eucalyptus. (Mention of commercial products or organizations in this paper does not imply endorsement by NIST.)

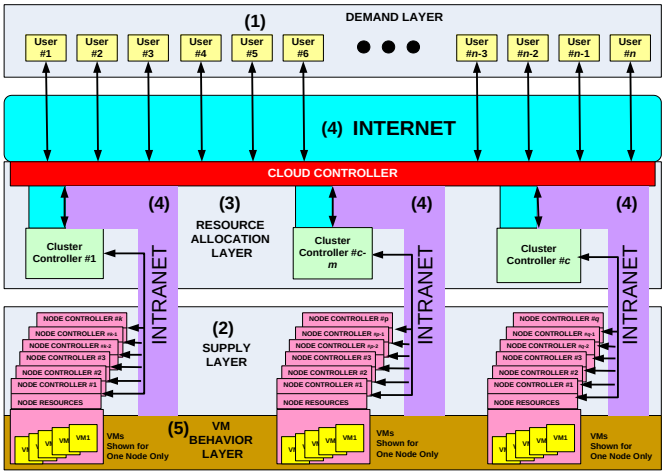


Figure 2. Schematic of Koala IaaS cloud simulator.

All nodes are placed geographically in a coordinate system of sites, where the cloud elements may be placed on one or multiple sites. User sites are selected randomly upon each user's arrival; user types are also assigned randomly. User type determines the quantity and mix of VMs the user will request, which can include a minimum number required to start an application and a maximum that can be exploited should sufficient resources be available. A user requests VMs, and the cloud controller can honor the request fully (maximum requested VMs) or partially (at least minimum requested VMs). If insufficient resources exist, the cloud responds with NERA (not enough resources available). Upon receiving a NERA response, a user may retry intermittently during a day, and if VMs cannot be obtained, then retire for the evening and return the next day to try again. After passing too many days without obtaining the needed VMs, the user gives up and leaves the system, only to be regenerated as a new user.

Upon successfully obtaining VMs, a user selects a holding time, during which VMs may be added or terminated. Of course, VMs may also fail within the cloud, so a user will attempt to maintain a required minimum number of VMs by requesting additional VMs as needed. When holding time expires, a user requests termination of all VMs and reenters the system as a new user.

The cloud controller handles all user requests, checking with subordinate cluster controllers to find available space for a collection of VMs, which are mapped to a single cluster to localize inter-VM communication. The cloud controller uses one of several algorithms [27] to select a specific cluster. Cluster controllers monitor the state of subordinate nodes, and use one of several algorithms [27] to select specific nodes for

 ISBN: 978-1-61208-308-7

placement (or relocation) of individual VMs. Under guidance, from a simulated administrator, the cloud controller can add and remove clusters from the cloud, and cluster controllers can add and remove nodes from clusters. The administrator can also terminate VMs that the cluster controller is unable to stop.

Table I categorizes 129 Koala parameters over which we conducted a GA search. More than half the parameters define element behaviors, most by the user and cloud controller, while 22 describe structural elements, half related to the network. The model also simulates failures that could occur in the network and among the physical platforms and components. A smaller set of parameters can inject behavioral and structural asymmetries, such as changing user demand profiles over time and allowing clouds to be constructed as a combination of large and small clusters, rather than of same-sized clusters.

TABLE I. SUMMARY OF KOALA PARAMETERS TO SEARCH OVER.

Model Element	Parameter Category				
	Behavior	Structure	Asymmetry	Failure	Total
User	28	2	4	0	34
Cloud Controller	21	4	5	0	30
Cluster Controllers	11	5	3	0	19
Nodes	6	0	0	14	20
Intra-Net/Inter-Net	4	11	2	9	26
Totals	70	22	14	23	129

Among the 129 parameters, we included four Booleans to turn on/off behaviors that control orphan VMs, a potential problem uncovered in earlier experiments with Koala [27], where lost messages could leave users and the cloud controller unaware that VMs had been allocated, leading to retries, reallocations, and ultimately to saturation of cloud resources. Most orphan VMs arise during initial allocation, but some are caused by failed terminations. Additionally, orphan VMs may occur as collections of VMs are relocated before a cluster is shut down. Logic was included in Koala to detect and remove orphans in all three classes, and an administrator was also simulated to allow residual orphans to be stopped manually.

When a cloud is saturated with VM orphans, users are unable to obtain requested VMs and eventually give up after exhausting their patience. For our GA search, we defined anti-fitness as the ratio of users who give up to the total number of arriving users. The larger the ratio, the more users were turned away, and the lower the cloud's revenue.

To guide the GA search, we defined a range and precision for each Koala parameter (see Table II for an elided list), yielding a mean of about six values per parameter, and thus a search space of approximately 10^{100} parameter combinations. Using our description, the GA computed the number of values (and bits) needed to encode a Koala parameter combination (as a binary string), and then randomly placed the binary encoding for each parameter into a 334-bit chromosome, which served as the internal form used by the GA to represent Koala parameters. The GA also provided routines to generate Koala parameter values from binary encodings given in chromosome form.

We deployed a distributed population of 200 Koala simulators on a high-performance cluster, under GA control (see Figure 3) via signal files in a shared, network file store. Each simulator, allocated to one core, waits for the GA to signal that a parameter file exists and then runs a simulation, reports the resulting anti-fitness value, and awaits the next signal. The GA generates parameter files for each simulator and periodically checks progress and collects anti-fitness reports as runs complete. Once all runs in a generation finish, the GA uses the algorithm described in Section III to create the next generation of parameter files, and so on until completing a specified number of generations.

TABLE II. MAPPING OF KOALA PARAMETERS TO CHROMOSOMES.

	Koala Parameter Space (Size = 10^{100})			Genetic Algorithm Computed Chromosome Map (Size = 2^{334})			
PARAMETER	MIN	MAX	PRECISION	#VALUES	LOW_BIT	HIGH_BIT	#BITS
P_CreateOrphanControlOn	0	1	1	2	36	36	1
P_TerminationOrphanControlOn	0	1	1	2	58	58	1
P_RelocationOrphanControlOn	0	1	1	2	11	11	1
P_AdministratorActive	0	1	1	2	330	330	1
P_clusterAllocationAlgorithm	0	5	1	6	31	33	3
P_describeResourcesInterval	600	3600	600	6	81	83	3
P_nodeResponseTimeout	30	90	30	3	210	211	2
P_TerminatedInstancesBackOffThreshold	3	6	1	4	56	57	2
P_TerminationBackOffInterval	180	360	60	4	88	89	2
P_TerminationRetryPeriod	600	1200	300	3	316	317	2
P_StaleShadowAllocationPurgeInterval	600	3600	600	6	242	244	3
P_cloudAllocationCriteria	0	3	1	4	321	322	2
P_clusterShadowPurgeLimit	1	21	5	5	290	292	3
P_instancePurgeDelay	180	600	60	8	98	100	3
P_clusterEvaluationResponseTimeout	60	120	30	3	14	15	2
P_MaxPendingRequests	1	10	1	10	72	75	4
P_CloudTerminatedInstancesBackOffThreshold	3	6	1	4	169	170	2
P_CloudTerminationBackOffInterval	180	360	60	4	40	41	2
P_CloudTerminationRetryPeriod	3600	10800	1800	5	297	299	3
P_ClusterShutdownGracePeriod	86400	2.59E+05	43200	5	147	149	3
●	●	●	●	●	●	●	●
P_RequestEvaluatorTimeoutWaitProportion	0.1	0.4	0.1	4	145	146	2
P_RequestEvaluatorClusterMinimumResponse	0.6	0.9	0.1	3	269	270	2
P_MaxRelocationDuratonProportion	0.65	0.95	0.1	4	90	91	2
P_MaximumRelocateDescribeRetries	4	16	2	7	254	256	3
P_AverageCloudAdministratorAttentionLatency	28800	86400	14400	5	308	310	3
P_AverageCloudAdministratorShutdownDelay	300	900	300	3	45	46	2
P_avgTimeToClusterCommunicationCut	2.88E+06	2.88E+07	2.88E+06	10	217	220	4

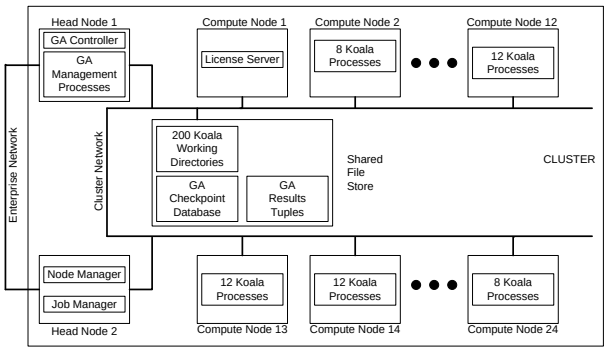


Figure 3. Schematic of simulators deployed on a cluster under GA control.

TABLE III. SETTINGS FOR KEY GA CONTROL PARAMETERS.

Generations	500
Population Size	200 Individuals
Elite Per Generation	16 Individuals
Reboot After	200 Generations
Selection Method	Stochastic Uniform Sampling
# Crossover Points	3
Mutation Rate	0.001 ≤ Adaptive ≤ 0.01

Table III shows settings we assigned for key GA control parameters. Mutation rate is controlled by an adaptive algorithm that increases mutation probability (and variance among parameter combinations) as the range of population fitness narrows and lowers probability upon divergence.

 ISBN: 978-1-61208-308-7

V. RESULTS AND DISCUSSION

To assess dynamics, quality, effectiveness, and cost of GA search, we steered a population of 200 Koala simulators over 500 generations, expecting the previously known VM orphan problem to be revealed in cases where Koala's orphan-control logic was disabled. Figure 4 shows three plots, where the *y*-axis gives (a) average, (b) standard deviation and (c) maximum anti-fitness vs. time (increasing generations). Mean anti-fitness starts low (around 0.2) for randomly generated parameter combinations, and peaks (at 0.79) within 11 generations, before falling to around 0.65 (2/3 of users not served) until generation 201 (also 401), when the GA randomizes parameters for the 184 non-elite individuals. After these reboots, mean anti-fitness rises quickly to over 0.7 and then falls back to around 0.65. Our shared file store suffered a disk crash, which required us to restart generation 311, using checkpoint information we save during the GA search. The restart caused a spike in mean anti-fitness, before settling back to around 0.65.

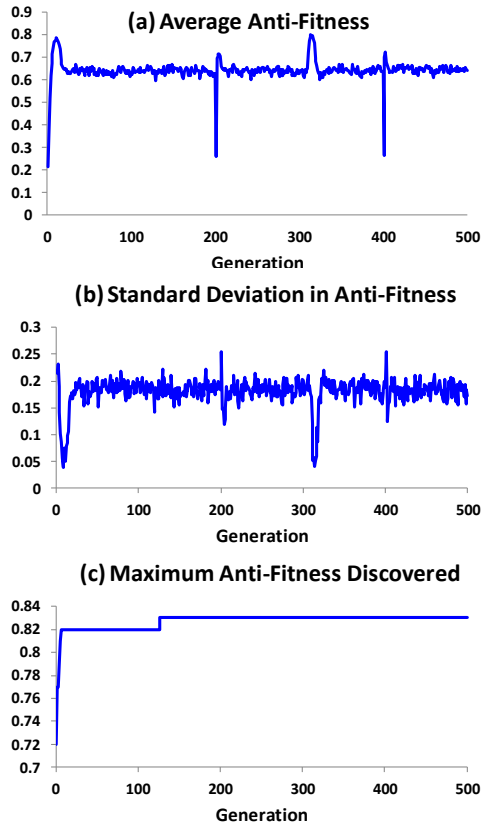


Figure 4. GA search dynamics in anti-fitness (*y*-axis)–(a) average, (b) standard deviation and (c) maximum–over 500 generations (*x*-axis).

The plot of standard deviation in anti-fitness inversely mirrors the average, i.e., high averages indicate low variance. As described previously, changes in the anti-fitness variance in a population stimulate automatic adjustments in mutation rate. The plot of maximum fitness shows that by generation 7 the GA had discovered scenarios where 82% of users could not be served, and by generation 127 the GA found scenarios where the proportion of non-served users increased to 83%. These results suggest that, for the Koala simulator, GA search could uncover failure scenarios within 100-200 generations.

Figure 5 gives a frequency distribution of anti-fitness values obtained for the (200 individuals × 500 generations =) 10^5 scenarios explored by the GA, which represent only a tiny fraction of the 10^{100} possible Koala scenarios. The histogram reveals that 84% of the scenarios explored by the GA yielded anti-fitness ≥ 0.50, despite the likelihood that most of the Koala search space consists of scenarios with low anti-fitness, as shown by the fact that randomly generated scenarios yielded mean anti-fitness of 0.2. Further, only 8.12% of the scenarios explored by the GA were duplicates, which is only slightly larger than the 8% elite individuals carried unchanged from generation to generation. These results indicate that the GA search explored predominantly non-duplicative scenarios with high anti-fitness.

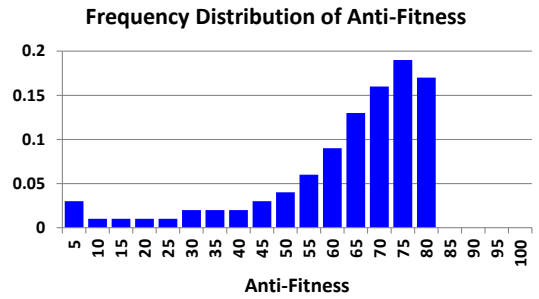


Figure 5. Histogram of anti-fitness values for all 10^5 parameter combinations.

Various analysis methods, such as feature extraction and clustering, may yield insights into failure causes. Here, we use differential probability analysis; comparing the estimated probability of each parameter-value (PV) pair appearing in scenarios with high anti-fitness against estimated probability of the same pair appearing in scenarios with low anti-fitness. We postpone, for future work, using additional analysis methods.

Let C be the set of collected tuples (recall Figure 1), each containing a vector of PV pairs and a corresponding anti-fitness value, f. We segmented C into high-pass (H) and low-pass (L) subsets: $H = \{x \in C \mid f_x > 0.70\}$ and $L = \{x \in C \mid f_x < 0.15\}$. For each PV in the high-pass subset, we estimated the probability of occurrence, $P(PV_i \mid f > 0.70)$, using the ratio $|PV_i \in H| / |H|$, representing the count of PV_i in the high-pass subset divided by the subset cardinality. We computed a similar estimate, $P(PV_i \mid f < 0.15)$, for each PV in the low-pass subset. Subsequently, we took the difference between the two estimates, $D = P(PV_i \mid f > 0.70) - P(PV_i \mid f < 0.15)$. A large positive difference suggests that a PV pair contributes to a failure scenario, while a large negative difference suggests that a PV pair contributes to desirable system behavior. Figure 6 plots D for 684 PV pairs, sorted by decreasing D, found in our GA search for a known failure scenario. We label significant outliers.

Figure 6 illustrates that most PV pairs exert little influence on failure or success scenarios, appearing about as often in both the H and L subsets. Six PV pairs appear to drive failure scenarios, and one PV pair shows most influence on success. The largest positive difference (0.58) occurs in the absence of logic to control orphans during initial VM allocation, while the largest negative difference (-0.58) occurs when that logic is present. In effect, this is the known failure scenario that we were expecting the GA to find. The second highest positive

 ISBN: 978-1-61208-308-7

difference (0.42) occurs when users select random request timeouts with an average of 30 s. By not waiting long enough for responses from the cloud, users create *virtual* message losses, because the receiving process has terminated before a response arrives. Without orphan-control procedures running, lost messages lead to a buildup of orphan VMs, leaving few resources available to serve users. This combined effect of short user timeouts and lack of orphan-control procedures was previously unknown to us. From these results, a designer might deduce that orphan-control procedures are needed, and that the cloud must find some means to ensure clients wait long-enough for the cloud to respond to requests.

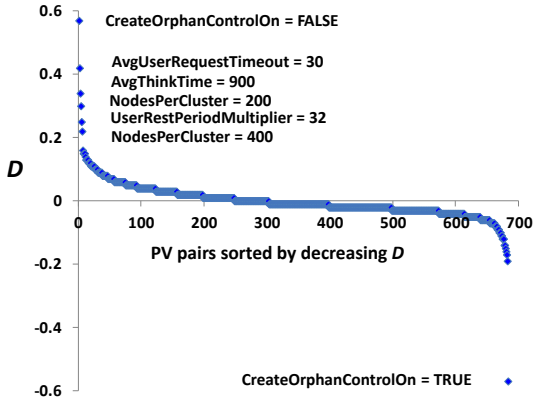


Figure 6. *D* (*y*-axis) for 684 sorted PV pairs (*x*-axis) for first GA search–outlier PV pairs labeled.

From the data in Figure 6, we were also able to identify two other potential failure scenarios: (1) cloud overload and (2) impatient users. When average cluster sizes were small (either 200 or 400 nodes), the cloud had insufficient resources to serve user demand. When the average user rest period (think time × rest period multiplier) was 8 hours, users tended to retry more frequently, and thus to give up in a shorter overall time.

A. *Costs of GA Search*

GA search for failure scenarios incurred costs of two types. First, substantial programming effort was required prior to the search. Second, GA search of simulation models can incur significant latency. We discuss each type of cost in turn.

Although the Koala simulator had been used for several years and executed robustly under diverse parameter settings, generating an initial population of random parameter combinations led to many crashes due to execution paths that were not previously encountered. Finding and fixing these software errors required significant effort. Further, the Koala simulator typically executes for a specified simulated time. The associated wall-clock time can vary widely depending upon the specific parameter settings used. To ensure deterministic search time, we modified Koala to terminate a simulation when either simulated time expired or a predetermined allocation of wall-clock time was reached. Though this was a relatively simple change, the Koala simulator had not been coded with the expectation that simulations could terminate from any given dynamic system state. Subsequently, abrupt terminations revealed many more simulator crashes, which had to be diagnosed and fixed.

Even after the Koala simulator was made sufficiently robust, numerous issues arose regarding the use of a cluster for executing simulator populations. Upon node failure, the cluster would restart simulators on some other available node. When the entire cluster failed and restarted, race conditions ensued among various components. Diagnosing the state of the entire simulator population proved difficult when using only available cluster and node management tools. To resolve such issues, it required significant effort to create a robust management system to control the population of simulators.

Executing a GA search can require substantial latency because all simulators in a given generation must complete before a next generation can be constructed. For our experiments, we limited each simulation to use no more than 90 minutes, which meant that we could complete 500 generations in 30 days. Our results showed, however, that for the Koala model we could generate failure scenarios within 100-200 generations. For that reason, we limited subsequent GA search iterations to about 200 generations, which typically complete within 14-16 days. These latency computations assume sufficient processors (one per simulator) are available for use over the entire search. If fewer processors are available, the search can take longer, though often shorter simulations can complete on a sequentially shared processor, while longer simulations execute on other processors. We completed iterative GA searches of 500, 205, 209, and 205 generations, which required a total of 74 days. These latencies suggest that GA search should be pursued only for systems with sufficient development time, and where failure scenarios have high cost.

B. *Additional Iterations of GA Search*

We conducted a second GA search; this time ensuring that orphan-control procedures and the cloud administrator were always active. Our goal was to evaluate the ability of GA search to find additional failure scenarios. We executed only 205 generations. Figure 7 plots estimated probability differences for the 677 PV pairs found by the GA.

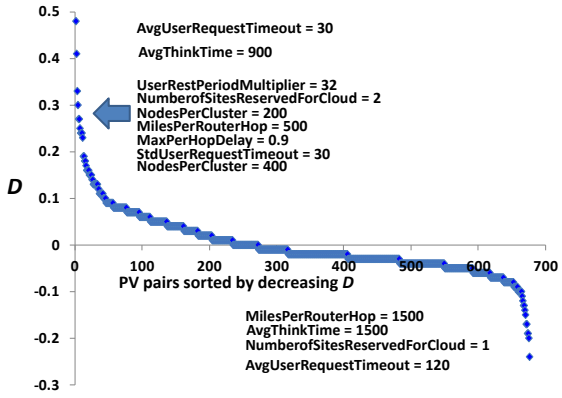


Figure 7. *D* (*y*-axis) for 677 sorted PV pairs (*x*-axis) for second GA search–outlier PV pairs labeled.

The largest positive difference (0.48) occurs in Figure 7 when the average user request timeout is 30 s. This is the same result found previously. This implies that if user timeouts are too short, then even with orphan-control procedures active, the cloud will fail to serve enough users, as the maximum anti-fitness was still 0.82, though the average decreased to about

 ISBN: 978-1-61208-308-7

0.55. Since orphan-control procedures operate over periods numbered in hours, virtual message losses caused by short user timeouts can still overtax the procedures. This finding was unknown previously. Another set of related parameters also exhibited large positive differences. For example, small standard deviations in user request timeouts tended to keep short user timeouts as short as possible. Short timeouts were exacerbated by increases in inter-site distances, especially when combined with short inter-router distances (i.e., more network hops between sites) and with higher simulated per-hop queuing delays. This implies that cloud designers must take wide latitude in considering many factors beyond their control that could determine the best user timeouts to encourage. On that issue, the PV with the largest negative difference (-0.24) was the user request timeout set to 120 s. This result implies that GA search can recommend optimal settings while simultaneously searching for failure scenarios. Finally, the iterated GA search also reestablished that small clusters would lead to overload and that impatient users could be a problem.

We conducted a third GA search over 209 generations. In that search, we changed the ranges of some parameter values to seek new failure scenarios and additional insights into system behavior. As expected, since we were searching for failure scenarios, the GA search found only slightly improved outcomes, yielding a maximum anti-fitness of 0.77 and an average of about 0.6. On the other hand, new insights were revealed. Figure 8 plots estimated probability differences for the 680 PV pairs found by the GA.

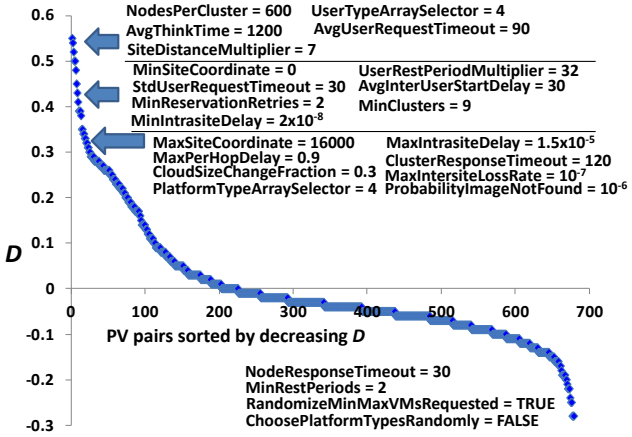


Figure 8. *D* (*y*-axis) for 680 sorted PV pairs (*x*-axis) for third GA search–outlier PV pairs labeled.

Though we increased (by 60 s) the range of user request timeouts, the new minimum timeout of 90 s proved to be too short, especially when coupled with specific network factors (such as long distances and large per-hop delays), along with 120 s delays by the cloud controller, when awaiting responses from clusters. Regarding response delays within the three-tiered cloud system, when the cloud waited only 60 s for cluster responses and clusters waited only 30 s for node responses, the system exhibited better outcomes. The user request timeout must accommodate delays due to network factors and timeouts within the cloud itself. The GA search found that an average user request timeout of at least 120 s (borderline outlier) was required to lower anti-fitness, and that 180 s (borderline outlier) provided the lowest anti-fitness.

Though we increased (by 400 nodes) the range of cluster sizes, a 600-node minimum size still proved too small. The GA found that at least 800 nodes per cluster were needed to avoid cloud overload for the parameters within the search space. Further, the GA discovered that a 30 s average inter-user startup delay, a parameter intended to gradually introduce load into the cloud, was too short, leading to cloud overload.

The GA found that homogeneous cluster sizes lowered anti-fitness, when compared with cases where 20% of clusters were large and 80% small. The GA also found that increasing and decreasing cloud size by 30% yielded higher anti-fitness than smaller size changes of 10% and 20%. Further, the GA found that cloud administrators needed to complete individual operations in a mean of 300 s (borderline outlier); 900 s (borderline outlier) was too long.

The GA also found insights related to platform types. First, assigning platform types randomly from a specified set (simulating a cloud constructed by adding any available nodes) increased anti-fitness. Second, one specific arrangement of platform types, where 28% of nodes had 32-bit architectures, increased anti-fitness when combined with simulated user types (60% web-service and 40% distributed-search applications) that required 64-bit architectures for all VMs.

All searches described above had the property that *H* subsets contained over 10^4 tuples, while comparable *L* subsets contained fewer than 10^3 tuples. This discrepancy in samples occurred naturally because the GA was searching for scenarios more likely to fall into *H* subsets. *L* subsets had as many as hundreds of tuples only because, as discussed previously, the low anti-fitness landscape of the Koala simulator was much larger than the high anti-fitness landscape. One could increase samples in *L* subsets by inverting the GA search to look for scenarios with low anti-fitness.

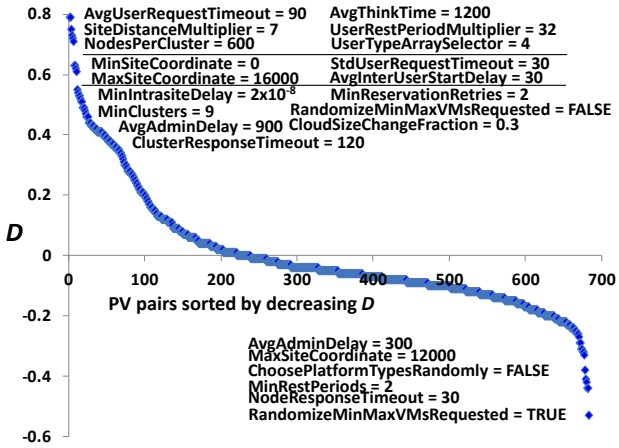


Figure 9. *D* (*y*-axis) for 683 sorted PV pairs (*x*-axis) for fourth GA search–outlier PV pairs labeled.

We inverted a fourth GA search. We used the same parameter space as in the third search, but instructed the GA to seek high fitness (i.e., low anti-fitness) scenarios. We ran the inverted search for 205 generations, and then combined the collected tuples with the tuples collected during the third search. After filtering, the resulting *H* subset contained 14601 tuples and the *L* subset contained 42253 tuples. Analysis of the

 ISBN: 978-1-61208-308-7

probability differences, shown in Figure 9, confirmed findings obtained from the third GA search.

VI. CONCLUSIONS AND FUTURE WORK

We defined a design-time method, combing GA search with simulation, to seek failure scenarios in system models. We applied the method in a case study, seeking (and finding) a known failure scenario in an existing IaaS cloud simulator. We iterated the GA search to reveal previously unknown failure scenarios. We used the case study to evaluate the dynamics, quality, effectiveness, and cost of GA search. Our GA searches explored predominantly non-duplicative scenarios with high anti-fitness. We uncovered evidence that GA search could reveal insights about optimal parameter settings, while simultaneously searching for failure scenarios. We also found that, due to high latency, GA search should be pursued only for systems with sufficient schedule time, and where failure scenarios have high cost.

We can extend our work in five directions. First, additional analysis methods need to be explored, to further mine the data collected by our GA searches. We can envision using statistical and information-theoretic techniques to extract features from the collected tuples, and then applying clustering algorithms to suggest specific classes of failure scenarios. Second, we should continue to explore our case study, attempting to uncover parameter subspaces where no failure scenarios can be found, and also using GA search under alternate definitions of anti-fitness to discover other kinds of system failure scenarios that might exist. Third, we should apply our method to models of other complex information systems, such as communication networks and other forms of computational clouds. This would allow us to confirm the generality of our approach. Fourth, we should seek partners, operating cloud computing systems or test beds, against which we can validate our method. Finally, we should investigate run-time methods to provide early signals of incipient failures. Such run-time methods are necessary because design-time methods are unlikely to discover all possible failure scenarios that could arise in a deployed system.

REFERENCES

[1] D. Takahashi, "Amazon's outage in third day: debate over cloud computing's future begins", VB/News, April 23, 2011.

[2] Z. Michalewicz and D. Fogel, How to Solve It: Modern Heuristics, Springer, 2nd ed., 2004.

[3] D. Fogel (ed), Evolutionary computation: the fossil record, IEEE Press, 1998.

[4] M. Mitchell, An introduction to genetic algorithms, MIT Press, 1998.

[5] M. Fischer and J. Shortle, "Rare event simulation: enhancing efficiency, SigmaΣ noblis, 10:1, Sept. 2011, p. 52.

[6] C. Kelling and G. Hommel, "Rare event simulation with an adaptive "RESTART" method in a Petri net modeling environment", Proceedings of the 4th WPDRTS, IEEE, Jan. 1996, pp. 229-235

[7] P. Ecuyer and B. Tuffin, "Splitting for rare-event simulation", Proceedings of the Winter Simulation Conference, IEEE, Dec. 2006, pp. 137-148.

[8] D. Reijsbergen, P-T de Boer, W. Scheinhardt, and B. Haverkort, "Rare event simulation for highly dependable systems with fast repairs", Performance Evaluation, 69:7-8, 2010, pp. 336-355.

[9] G. Galati, M. Naldi, and G. Pavan, "Stochastic simulation techniques as related to innovation in communications-navigation-surveillance and air traffic management", Simulation Modeling Practice and Theory, 11, 2003, pp. 197-209.

[10] A. Shultz, J. Grefenstette, and K. DeJong, "Learning to break things: adaptive testing of intelligent controllers", Handbook of Evolutionary Computation, Chapter G3.5, IOP Publishing Ltd and Oxford University Press, 1995, pp. 1-11.

[11] E. Yucesan and S. Jacobson, "Computational issues for accessibility in discrete event simulation", ACM Transactions on Modeling and Computing Simulation, 6:1, 1996, pp. 53-75.

[12] A. Haines, K. Mills, and J Filliben, "Determining relative importance and best settings for genetic algorithm control parameters", NIST Pub. #912472, Nov. 2012, pp 1-22.

[13] C. Dabrowski and F. Hunt, "Using Markov chains and graph theory concepts to analyze behavior in complex distributed systems", Proceedings of the 23rd European Modeling and Simulation Symposium, Sept. 2011, pp. 1-10.

[14] G. Fainekos, S. Sankaranarayanan, K Ueda, and H. Yazarel, "Verification of automotive control applications using S-TaLiRo," American Control Conference (ACC), Jun. 2012, pp. 3567-3572.

[15] Y. Matsuo, "Prediction, forecasting, and chance discovery", Chapter 3 in Chance Discovery, Springer, 2003, pp. 30-42.

[16] F. Salfner, M. Lenk, and M. Malek, "A survey of online failure prediction methods", ACM Computing Surveys, 42:3, 2010, Article 10, pp. 1-42.

[17] P. Gross et al., "Predicting electricity distribution feeder failures using machine learning susceptibility analysis", Proceedings of the National Conference on Artificial Intelligence, 21:2, MIT Press, Jul. 2006, pp. 1705-1711.

[18] Q. Guan, Z. Zhang, and S. Fu, "Proactive failure management by integrated unsupervised and semi-supervised learning for dependable cloud systems", 6th International Conference on Availability, Reliability and Security, Aug. 2011, pp. 83-90.

[19] F. Salfner, "Predicting failures with hidden Markov models", Proceedings of 5th European Dependable Computing Conference, Apr. 2005, pp. 41-46.

[20] G. Weiss and H. Hirsh, "Learning to predict extremely rare events", Proceedings of the AAAI Workshop on Learning from Imbalanced Data Sets, Jul. 2000, pp. 64-68.

[21] Q. Guan, Z. Zhang, and S. Fu. "Ensemble of bayesian predictors for autonomic failure management in cloud computing", Proceedings of IEEE International Conference on Computer Communications and Networks, Jul. 2011, pp. 1-6.

[22] Y. Watanabe, H. Otsuka, M. Sonoda, S. Kikuchi, and Y. Matsumoto, "Online failure prediction in cloud datacenters by real-time message pattern learning", IEEE 4th International Conference on Cloud Computing Technology and Science, Dec. 2012, pp. 504-511.

[23] T. Chalermarrewong, T. Achalakul, and S. Wee See, "Failure prediction of data centers using time series and fault tree analysis", Proceedings of the IEEE 18th International Conference on Parallel and Distributed Systems, Dec. 2012, pp.794-799.

[24] F. Salfner and P. Tröger, "Predicting cloud failures based on anomaly signal spreading", 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Jun. 2012, pp. 1-2.

[25] C. Dabrowski and K. Mills, "VM leakage and orphan control in open-source clouds", Proceedings of IEEE CloudCom, Dec. 2011, pp. 554-559.

[26] K. Mills, J. Filliben, and C. Dabrowski, "An efficient sensitivity analysis method for large cloud simulations", Proceedings of the 4th International Cloud Computing Conference, IEEE, Jul. 2011, pp. 1-8.

[27] K. Mills, J. Filliben, and C. Dabrowski, "Comparing VM-placement algorithms for on-demand clouds", Proceedings of IEEE CloudCom, Dec. 2011, pp. 91-98.

 ISBN: 978-1-61208-308-7

A Matlab/Simulink Simulation Approach for Early Field-Programmable Gate Array Hardware Evaluation

Celso Coslop Barbante, José Raimundo de Oliveira
Computing Laboratory (COMLAB)
Department of Computer Engineering and Industrial Automation (DCA), UNICAMP
Campinas, Brazil
e-mail: celsocos@dca.fee.unicamp.br, jro@dca.fee.unicamp.br

***Abstract*— This paper presents a Matlab test bench development for Field-Programmable Gate Array hardware simulation. When a design uses hardware blocks provided by third-part vendors (known as Integration Packages - IP), several options can be set in the block configuration page, inside vendor tool, and affect how the block behaves. These configuration options should be evaluated for any integration package one may be interested in and the test bench proposed facilitates the evaluation of any block-specific configuration parameters, enabling a three times reduction of block configuration time.**

Keywords-Model verification; Matlab; FPGA design.

I. INTRODUCTION

Hardware verification is becoming more challenging as design complexity grows. Verification times have increased with the rising gate count; as overall design complexity grows, ensuring that the system complies with the required specification in early design stage is a desired time saving approach [1].

Textual language can be used to develop a test bench; however, this approach has a degree of complexity similar to design itself and is human-resource intensive. This task can be accomplished easier with a tool like Matlab, demanding less knowledge of vendor specific optimizations [2] to achieve the goal of developing a test bench.

The required computational run-time for simulations is also an important factor to consider because computer resources are limited and costly. Efficient simulation techniques, as presented in this work, collaborates to improve a rational use of computer resources [3].

According to a survey of Collett International Research in 2002, only 39% designs were shipped bug free at first silicon, while 60% contained logic or functional flaws, more than 20% required 3 or more silicon spins. The Collett survey has also shown that nearly 50% of total engineering time was spent on verification [4]. Because the design complexity continually increases, the actual numbers are expected to be worse, being more difficult to verify the design today than in 2002.

Some of these verification challenges can be addressed by using a model-based simulation system, where mathematical aspects and algorithms become a key area in the verification efforts, also incorporating software techniques for formal verification [5-6]. The design methodology that best fits the proposed test bench is a top-bottom design strategy, which can be done using the Matlab and the FPGA tools to generate desired IP models; for example, Fast Fourier Transforms, Arithmetic Logic Units and Decoders, among other blocks that may be provided by Field-Programmable Gate Array (FPGA) vendors.

During design cycles, the model is refined and progressively approaches the hardware behavior, until the hardware IP can be directly used.

In this article, the Xilinx and Matlab integration will be presented with one simple test case, which consists of a Fast Fourier Transform (FFT) processing core, as presented in Section II. A proposed Design Flow is explained in Section III, and the methodology results are presented in Section IV.

II. MATLAB/FPGA SYSTEM INTEGRATION

The first step to achieve the goal to simplify verification using the Matlab is selecting an IP, from available from FPGA vendors, sometimes through third parties companies, but with vendor's support in order to enable support for simulation, hardware models and even synthesis. Two examples of such integration tools are Altera DSP Builder [7] and Xilinx System Generator [8].

Both tools share the same principles, but differ in integration method, capability and support options. The installation procedure details can be found in the vendor's web site and will not be repeated here; however, the process is straightforward once you have checked the Matlab version and FPGA tool version compatibility [9].

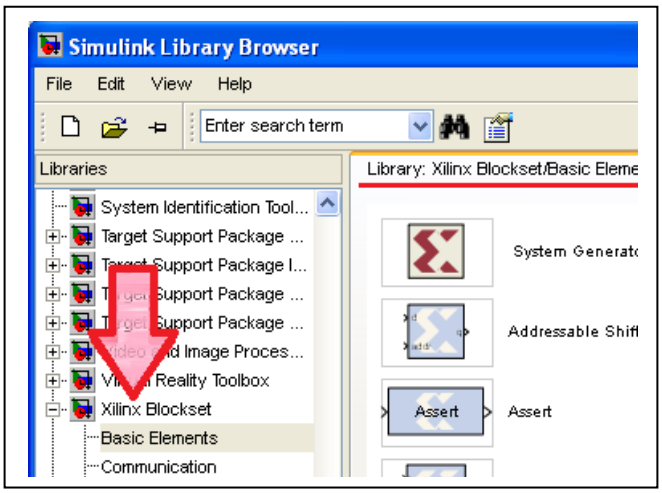


Figure 1. Xilinx Blockset inside Simulink

 ISBN: 978-1-61208-308-7

Each integration package is targeted to a specific Matlab/FPGA vendor tool combination and one must confirm you have correct version of all tools to avoid interoperability problems. The final result will be a Simulink block list inside Matlab, as it is shown in Figure 1.

Under the hood, more changes in the Matlab tool were done and more than a few Simulink [10] blocks are available: a diverse set of FPGA hardware models can be used and a clever use of this integration enable that a test bench developed in Matlab can also be used to validate hardware design, however to use the same Matlab environment for FPGA hardware models you must first generate the models inside the FPGA vendor tool.

During hardware model generation, design choices are required. These choices are made by parameters selections, and each IP has several parameters to be set. Early evaluation methodology to quickly test the model parameters is a major goal for this work, since several blocks can be created, and using the proposed test bench, parameters can be quickly adjusted and compared.

The Xilinx System Generator IPs available in functional categories are, in alphabetical order:

- Automotive & Industrial
- Advanced eXtensible Interface (AXI)
- Base IP (FPGA basic blocks)
- Communication and Networking
- Debug and Verification
- Virtual Input/Output
- Digital Signal Processing
- FPGA Features and Design
- Math Functions
- Memories and Storage Elements
- Standard Bus Interfaces
- Video and Image Processing

For instance, a single Fast Fourier Transform (FFT) core from Digital Signal Processing category should be somewhat simple to use, but this simple IP requires a lot of design choices: The FFT Core can compute from 8 to 65536-point forward or inverse complex transforms. The input data is represented as two's-complement numbers from 8 to 34 bits wide or single precision floating point numbers with 32 bits wide and the phase factors can range from 8 to 34 bits wide.

The FFT IP can use on-chip block RAM or distributed RAM across FPGA; calculation can be done using full-precision unscaled numbers, scaled fixed-point numbers and block-floating point. Some parameters can be configured in run time with additional logic: the point size, the choice of forward or inverse transform, and the scaling schedule.

Finally, four architectures are available to provide a tradeoff between sizes and transform time [11]. With all these options, just for a FFT block, the time required to simulate a hardware level design can be too long, so using hardware models with regular Matlab script files can easier simulate the generated IP.

All major parameters in FFT block generation can be seen in Figure 2, and all this options can be exercised in the proposed test bench.

The test bench is reliable due to modular nature: test cases and model files are separated, thus easier than other graphical-based tools to find issues in the test bench, simulation test cases and add supporting functions. Early simulation in the design flow, before first synthesis, can be used for exercise several design options ahead going further with the project.

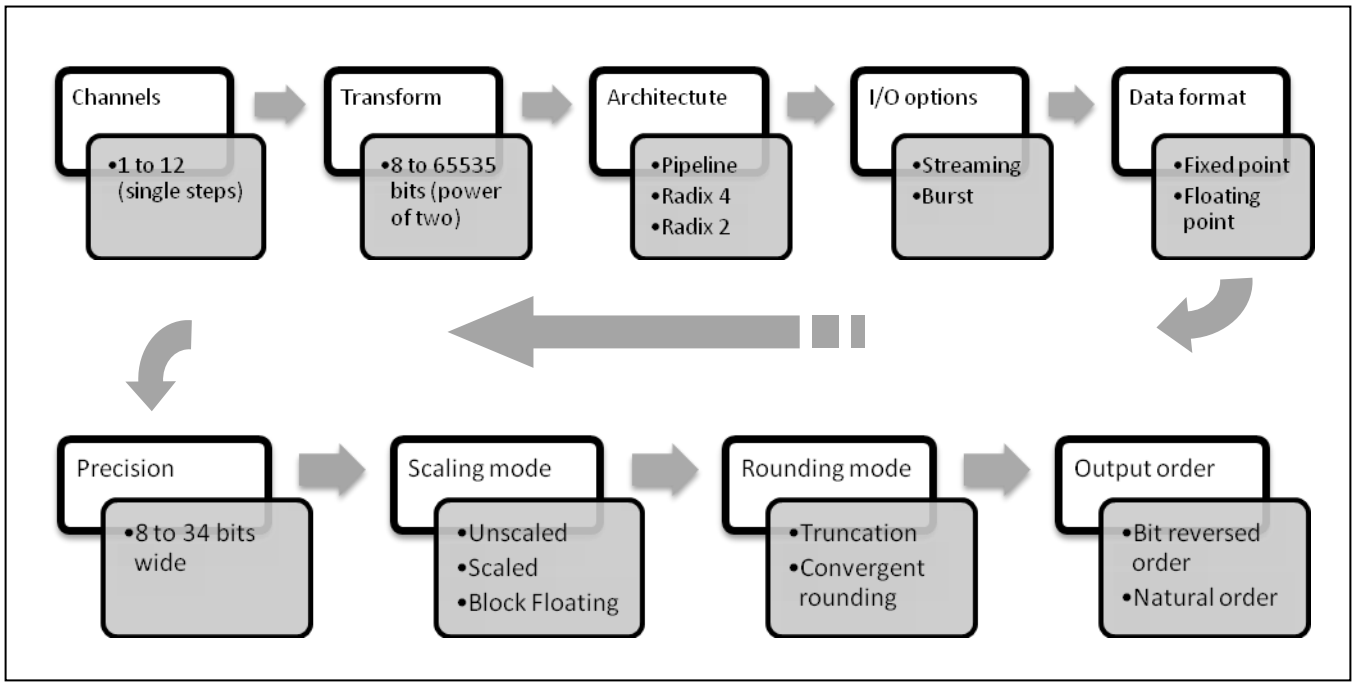


Figure 2. Main options for FFT model generation using Xilinx ISE Core Generator

 ISBN: 978-1-61208-308-7

III. MATLAB/FPGA DESIGN FLOW

A. Developing the matlab test bench

The Matlab is well known as a high-level language and interactive environment for numerical computation, visualization and programming [12]. New features like Hardware Description Language Coder (HDL Coder) and Hardware Description Language Verifier (HDL Verifier) allow modeling, simulating and exploring algorithms by Matlab and Simulink tools and FPGA vendor companion allows generating either target-independent or target-optimized hardware code and program Xilinx and Altera FPGAs.

Figure 3 shows the proposed methodology with a Matlab software model in step one, a hardware model introduced in step two, before hardware verification and reusing Matlab scripts and Simulink diagrams, saving time and test bench code. The final step in the design flow is the real hardware simulation.

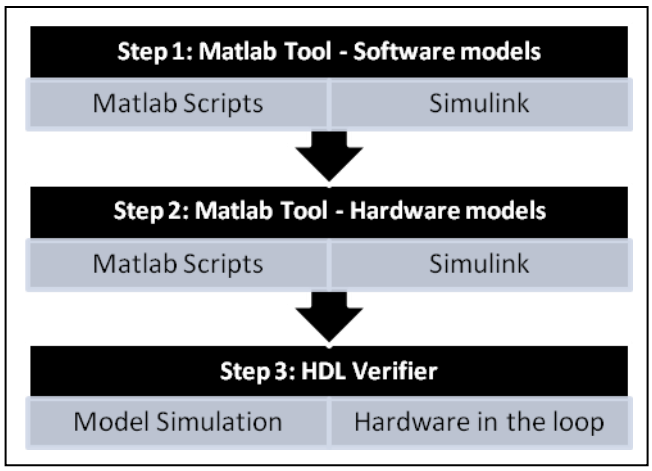


Figure 3. Proposed design flow

To check the FPGA design parameters against the system-level specifications a set of test cases must be developed. The initial test case could use only Matlab functions to model the system under study without adding hardware complexity, to validate the design idea before investing time in device selection, pin-out and others synthesis related issues. The initial test case is very simple and can easily simulate and compare the generated IP against Matlab floating point and full precision functions.

This allows design space analysis and numerical precision evaluation and also keeps the data for latter comparison between hardware and Matlab implementation, creating a figure of merit for quality and easily showing tradeoffs impact and artifacts that may arise due to several design choices made in IP generation.

The Xilinx provided C-model is cycle-accurate and has been demonstrated that results from Matlab, FFT model and System Generator model are all equivalent [13].

To reproduce the results of this work, please note you are not able to use the LCC compiler shipped with Matlab because it will not compile some IP models. Xilinx recommends using Microsoft Software Development Kit (SDK) for windows platform or Gnu Compiler Collection (GCC) for Linux platforms. You can refer to Xilinx user guide [14] and Matlab documentation [15] to create the function models whether is needed.

The test bench starts with a set of Matlab scripts that contain the global variables to control verbosity (to facilitate test bench debug) and design parameters. The test bench calls a set of test cases which can instantiate different models of device under test, for example the first one with the Matlab models and a second one with hardware models provided by FPGA tool.

B. File structure

A test bench top file was created with global variables to control run-time parameters, data sizes and script verbosity. There are parameters to controls text displayed messages during the test bench run and it is useful to debug the test bench itself, but once the environment is working, less debug messages can be displayed to concentrate the focus on the device under test. Utility file functions keep test bench organized and are a good location to place common functions brought by test cases. These are the test bench root files and a set of test cases files to exercise the model.

Once the main scripts and test cases were developed, a high level model using only built-in Matlab functions was made to mimic the desired hardware behavior. These high level models uses all the Matlab capabilities to reduce design time and the result will be compared to the hardware model and to latter validate the IP.

During the development of this work, a small set of utility routines were split into test bench top file and utility functions to keep top file small and easy to change. Moreover, the utility functions can be easily expanded and currently are used to display graphics and store personal preferences in a place which makes more sense than test bench top.

The hardware model can be instantiated by using the FPGA blocks available in Simulink but pay attention that hardware models can also be generated from FPGA vendor tool and embedded in Matlab code by using precompiled functions in the same fashion as regular function is used.

- Test bench top
  - control variables
  - data initialization
  - verbosity control
  - test case selection
- Utility functions
  - draw graphics
  - evaluate errors
  - formatted text output
  - other used test bench functions
- Test cases
  - call software and hardware models
  - execute desired tests and comparisons
  - use of utility functions and model files

 ISBN: 978-1-61208-308-7

- FFT Model direct calculation using Matlab function
  - golden model for floating point FFT
- FFT Model using fixed point calculation
  - golden model for fixed point FFT
- FFT Model using FPGA hardware model A
  - hardware model with "A" parameter set
- FFT Model using FPGA hardware model B
  - hardware model with "B" parameter set
- FFT Model using FPGA hardware model C
  - hardware model with "C" parameter set
- More FFT models can easily be included.
  - create as many models as required

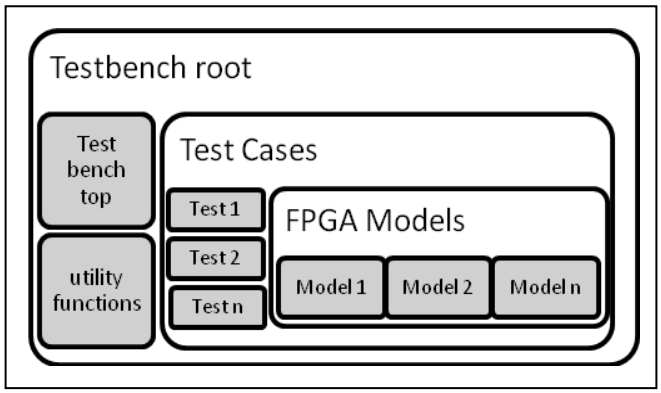


Figure 4. File structure example

When a FPGA IP model is generated by using the Core Generator tool, the model file is also generated. This model is placed inside IP directory tree, created by Core Generator tool and can be used in Matlab, but the integration between Matlab and FPGA vendor tool must be up to date because the Matlab model will call a pre-compiled file (for Altera) or bit accurate C models (for Xilinx).

Keep in mind that simply copying the model file for another PC running Matlab will not work, because the models rely on FPGA vendor files to work.

In the example provided in Figure 4, three files for FFT test cases and three files for FPGA models are shown, but during development, many files can be created as many options can be evaluated in the IP generation procedure. Use as many files as required to represent different parameters analyses in the test bench.

For a quick analysis of design space and numerical precision loss, this is very convenient, simple to design and to reuse.

IV. RESULTS

Once the test bench is ready and the software level function is working, the simulation is very simple to be repeated because the entire test bench is parameterized.

The Matlab scripts calculate buffer sizes and compare with provided data and other similar tasks, in order to provide a sanity check during simulation run time. This help to avoids mistakes in data format and parameter setup, because the sanity checks try to reproduce the constraints available in the FFT manual.

The result from this test bench development using automatic calculation instead of hard-coded values is a deeper knowledge of the FFT IP and deeper comprehension of the FFT IP manual.

The test bench functions generate warning messages, trigger some double checks in IP specification to confirm if test bench behavior was accurate.

It was possible to simulate the hardware and evaluate design tradeoffs, simply using the model and comparing the results between high level function and the vendor-provided hardware model function.

A lot of experimentations with FFT IP parameters were possible, helping to find the best fit for IP speed, area size and data size. All those experimentations were easy to reproduce by saving the hardware model files, reducing the amount of time compared with traditional Simulink design flow with lower level test benches. The easy reproducible results are welcome result of this methodology.

Compared with traditional Simulink graphical approach, this text-based methodology with multiple model files that can be exercised in the same simulation run, has potentials to give early-results for comparison, enabling for example the analysis of fixed point quantization errors early in design saving synthesis time and reducing efforts in final hardware creation [16-17].

For the FFT IP core generated with parameters presented in Figure 2, the result for simulation with hardware and software models is depicted below. The algorithm uses the FFT block to simulate a power spectrum calculation from a modified sinusoid signal with a peak power at low frequency.

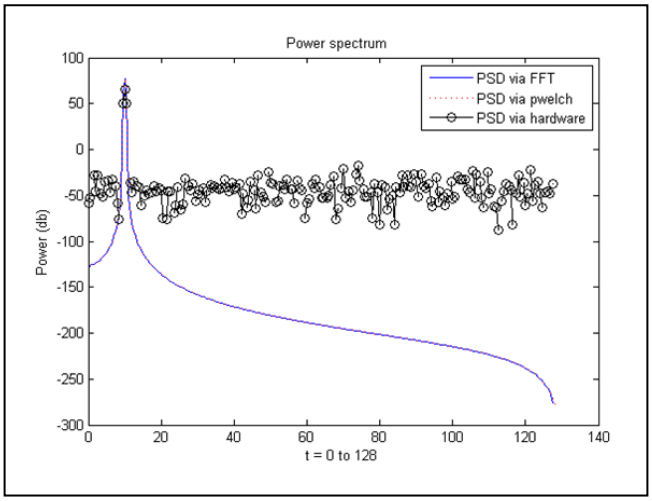


Figure 5. Power spectrun evaluation: Comparing pwelch function, software FFT model and hardware FFT model

Three simulations are shown in the Figure 5. A first one uses software function to calculate the power spectrum coefficients, the second uses Matlab pwelch function, and finally, the same calculation is done with FFT via hardware model.

It is possible to see that pwelch and Matlab FFT functions match to each other. This makes sense because

 ISBN: 978-1-61208-308-7

both functions use the same precision to calculate the power. However, a significant difference between the result from the Matlab software functions and the hardware model equivalent can be found.

This is expected because hardware models use fixed point precision and with the proposed test bench this result was easy to reproduce for several model configurations, resulting a better design space analysis.

The result provided in Figure 5 was found before first FPGA synthesis, directly in Matlab and much faster than traditional Simulink flow with hardware in the loop simulation.

V. CONCLUSIONS

A reduction of three times in simulation setup time was experienced for the FFT block. Saved time increases proportionally to quantity of simulated models, because once the file structure is deployed and first model is exercised, it is very simple to add more models to the test bench. The test bench development contributed to a better understanding of IP parameters, design arguments and options.

Initial hardware results were ready to analysis without even synthesize the designs and the Simulink graphical interface was avoided and this is an important feature, because it is faster to design a test bench by using text than graphical interface.

ACKNOWLEDGMENT

The authors would like to acknowledge the financial support of CAPES and the support of the Department of Computer Engineering and Industrial Automation (DCA) at State University of Campinas – UNICAMP.

REFERENCES

[1] E. Linehan and S. Clarke, “Managing hardware verification complexity with aspect-oriented model-driven engineering”, [retrieved: September, 2013], Available: http://ulir.ul.ie/handle/10344/666

[2] Synopsys Inc., “Faster simulation without more hardware”, [retrieved: October, 2013], Available: http://www.synopsys.com/Company/Publications/SynopsysInsight/Pages/Art2-FasterSimul-IssQ2-11.aspx

[3] S. Narayanan and L. Rothrock (editors), “Determining the number of simulation runs: Treating simulations as theories by not sampling their behavior”, Human-in-the-loop simulations: Methods and practice, Chapter 5, Springer, 2011, pp. 97-116

[4] P. J. Mosterman, “Model-based design of embedded systems”, Proceedings of the 2007 IEEE International Conference on Microelectronic Systems Education, IEEE Computer Society, IEEE Press, June, 2007, pp. 197-199, doi:10.1109/MSE.2007.65

[5] P. S. Kaliappan, “Model based verification techniques”, May, 2008, unpublished paper

[6] V. D. Silva, D. Kroening, and G. Weissenbacher, “A survey of automated techniques for formal software verification”, IEEE Transaction on Computer-Aided Design of Integrated Cricuits and Systems, Vol 27, no 7, July, 2008, pp. 1165-1178

[7] Altera Inc., “DSP builder”, [retrieved: August, 2013], Available: http://altera.com/products/software/products/dsp/dsp-builder.html

[8] Xilinx Inc., “System generator for DSP”, [retrieved: September, 2013], Available: http://www.xilinx.com/tools/sysgen.htm

[9] Xilinx Inc., “Which versions of System Generator for DSP and Accel DSP synthesis tool are compatible with which versions of ISE design tools and MATLAB?”, [retrieved: September, 2013], Available: http://xilinx.com/support/answers/17966.htm

[10] The Mathworks Inc., “Simulink - simulation and model based design”, [retrieved: October, 2013], Available: http://www.mathworks.com/products/simulink/

[11] Xilinx Inc., “Logic core IP: Fast fourrier transform product specification”, [retrieved: August, 2013], Available: http://www.xilinx.com/support/documentation/ip_documentation/ds808_xfft.pdf

[12] The Mathworks Inc., “Matlab (rel. 2011a – 2012b)”, [retrieved: September, 2013], Available: http://www.mathworks.com/

[13] J. Wu, “FFT results from Matlab fft, bit accurate C model and SysGen FFT block”, July, 2010, [retrieved: August, 2013], Available: http:// myfpgablog.blogspot.com.br/2010/07/fft-results-from-matlab-fft-bit.html

[14] Xilinx Inc., “LogiCORE IP fast fourier transform v9.0: product guide for vivado design suite”, [retrieved: August, 2013], Available: http://china.xilinx.com/support/documentation/ip_documentation/xfft/v9_0/pg109-xfft.pdf

[15] The Mathworks Inc., “Create MEX-files: Build C/C++ and Fortran subroutines into MATLAB functions” [retrieved: September, 2013], Available: http://www.mathworks.com/help/matlab/create-mex-files.html

[16] The Mathworks Inc., “FPGA design and codesign with Matlab”, [retrieved: August, 2013], Available: http://www.mathworks.com/fpga-design/

[17] S. V. Beek and S. Sharma, “Best practices for FPGA prototyping of MATLAB and simulink algorithms”, [retrieved: September, 2013], Available: http://www.eejournal.com/archives/articles/20110825-mathworks/

 ISBN: 978-1-61208-308-7

Rapid Weighted Random Selection in Agent-based Models of Infectious Disease Dynamics Using Augmented B-trees

Roel Bakker*‡, Tony Busker*, Richard G. White † and Sunil Choenni*
* Creating 010, Rotterdam University of Applied Sciences, Rotterdam, Netherlands
† Department of Infectious Disease Epidemiology, Faculty of Epidemiology and Population Health
London School of Hygiene and Tropical Medicine, London, UK
‡ Skardahl BV, Rotterdam, Netherlands
Email: r.bakker@hr.nl, a.l.j.busker@hr.nl, richard.white@lshtm.ac.uk, r.choenni@hr.nl

***Abstract*—Agent-based models (ABMs) are important tools for predicting infectious disease epidemics and for designing effective interventions. ABMs take into account individual differences, for instance in contact rate. The drawbacks of ABMs are high complexity and low performance. In this paper, we present a data structure - an augmented B-tree - to speed up the weighted random selection of individuals for the next transmission event in an ABM of infectious disease dynamics. An additional feature of the augmented B-tree is that it allows aggregating the force of infection for groups of simulated individuals. In short, our technique enhances the performance and simplifies the development of ABMs.**

***Keywords*—weighted random selection; ABM; agent-based modeling; infectious disease epidemics; B-tree; performance.**

I. INTRODUCTION

Agent-based models (ABMs) are important tools for predicting infectious diseases epidemics and for designing effective interventions [1]–[4].

The classic model for infectious disease dynamics is the so-called 'SIR model' formulated by Kermack and McKendrick [5] where S, I and R denote the susceptible, infected and recovered fractions of the population.

The original SIR model is a deterministic model where a set of differential equations describe the rates of change in S, I and R.

Stochastic models take both chance and the effect of population size into account, and model a population as discrete numbers of people in the S, I and R state. Each simulation run has a different outcome and evaluating a single scenario requires multiple runs and aggregating the output. These simulations take time as every single event (disease transmission, recovery) is modelled explicitly.

Agent-based models [6] are the most sophisticated type of model. In this type of model, individuals are represented as objects that differ from each other in the values of their attributes. In addition to taking chance and finite population size into account, agent-based models also take heterogeneity between individuals into account: some individuals may have higher contact rates than others, and some individuals may always recover from disease faster than others.

The main drawbacks of ABMs are high complexity and low performance. Each individual is a distinct object in software with its own attributes and life history. As in stochastic models, the time course of a simulation of an infectious disease with an ABM consists of a sequence of events (either transmission or recovery). If individuals differ in contact rate and/or recovery rate, a weighted random selection of individuals is required at each event. Selecting a single individual by iterating a list takes time proportional to the number of individuals in the list.

In this paper, we present an augmented B-tree as an efficient data structure for random selection of individuals weighted by the value of an individual attribute such as contact rate. The augmented B-tree is key for simulating epidemics in large (>100,000 individuals) populations with individual heterogeneity. The augmented B-tree can also be used to pinpoint the force of infection in a simulated population. In short, the data structure improves the performance of ABMs and makes it simpler to develop these models, which improves the tractability of this type of models.

The rest of this paper is laid out as follows. In Section II, we will provide the necessary background on the different types of SIR models. Section III describes the data structure in detail and reports performance figures. In Section IV, we will show results of using the data structure in an individual-based SIR model with different degrees in contact rate heterogeneity. Section V discusses the application of the data structure to an age structured population. In Section VI, we discuss additional features of this data structure and similar developments in the field of simulating networks of chemical reactions.

II. BACKGROUND ON INFECTIOUS DISEASES DYNAMICS

The classic model for infectious disease dynamics is the SIR model [5]. In this model, the population is subdivided into susceptible (S), infected (I) and recovered (R) categories. The following set of differential equations determines the dynamics:

$$dS/dt = -\beta \cdot c \cdot I \cdot S/N \quad (1)$$

$$dI/dt = \beta \cdot c \cdot I \cdot S/N - I/d \quad (2)$$

$$dR/dt = I/d \quad (3)$$

with

$$N = S + I + R \quad (4)$$

and

β	transmission probability per contact
c	contact rate
d	duration of infection

 ISBN: 978-1-61208-308-7

This model simulates numbers of individuals (or population fractions) as continuous variables and is deterministic: for a given set of initial values the model will always produce the same output. This type of model aims to capture the average behaviour of the epidemic.

Stochastic models take the random nature of transmission events between discrete individuals into account. A stochastic model simulates discrete numbers of people in the S, I or R state and produces different output for each simulation run. A stochastic equivalent of the deterministic SIR model simulates the transition events from the $S \rightarrow I$ state and the $I \rightarrow R$ state for discrete individuals, with:

$$r_{S \rightarrow I} = -dS/dt = \beta \cdot c \cdot I \cdot S/N \quad (5)$$

$$r_{I \rightarrow R} = dR/dt = I/d \quad (6)$$

The direct method [7] is an algorithm for stochastic models of chemical reaction kinetics that can be used to simulate the dynamics of this system:

1) sum the event rates
2) draw a random number x between 0 and the sum of the rates i.e., uniform on $(0, r_{S \rightarrow I} + r_{I \rightarrow R})$
3) determine whether infection or recovery will occur: infection if $0 < x < r_{S \rightarrow I}$ and recovery if $r_{S \rightarrow I} <= x < r_{S \rightarrow I} + r_{I \rightarrow R}$
4) draw a value for Δt from an exponential distribution with rate $r_{S \rightarrow I} + r_{I \rightarrow R}$
5) move the time forward to t+Δt and execute the event
6) go to step 1

Deterministic and stochastic models are relatively simple to implement although running stochastic models may be time consuming - especially for large populations - as every individual state transition is simulated.

Stochastic models do not take consistent heterogeneity between individuals into account. Stochastic models model numbers of molecules of a species or numbers of individuals in a certain state. Although it is possible to categorise a population into subgroups with different contact rates (e.g., the core group model for gonorrhea in the US [8]), an arbitrary and continuous distribution of contact rates (and/or recovery rates) within a population requires modelling at the level of the individual.

In the stochastic SIR model, the infection and recovery event rates are easily calculated from (5) and (6) and moving the simulation forward in time using the direct method is straightforward. In an agent-based SIR model with heterogeneity in both contact rate and duration of infection, the event rates are given by (assuming proportionate mixing):

$$r_{S \rightarrow I} = \beta \cdot \sum_{j=1}^{I} c_j \cdot \frac{\sum_{j=1}^{S} c_j}{\sum_{j=1}^{N} c_j} \quad (7)$$

$$r_{I \rightarrow R} = \sum_{j=1}^{I} \frac{1}{d_j} \quad (8)$$

In (7) the summed contact rate $\sum_{j=1}^{I} c_j$ of infected individuals replaces the product of contact rate and numbers of infected individuals $c \cdot I$ of(5). As we assume proportionate mixing, the summed contact rates of susceptible divided by the summed contact rate of all individuals $\sum_{j=1}^{S} c_j / \sum_{j=1}^{N} c_j$ in (7) replaces the fraction of contacts with susceptibles S/N of (5).

The principle of the direct method still works, but now we do not only have to determine which *type* of event occurs but also which *individual* should be selected for the transition event. Therefore step 5 in the algorithm is replaced by:

5) move the time forward to t+Δt and execute the event:
   a) if infection:
      - draw a random number y uniform on $(0, \sum_{j=1}^{S} c_j)$
      - iterate over all susceptibles subtracting c_j from y until $y < 0$
      - select that individual and execute the infection event
   b) if recovery:
      - draw a random number y uniform on $(0, \sum_{j=1}^{I} 1/d_j)$
      - iterate over all infected subtracting $1/d_j$ from y until $y < 0$
      - select that individual and execute the recovery event

The random selection of an individual weighted by contact rate (or recovery rate) in steps 5a and 5b performs poorly if individual rates are stored in a simple data structure such as an array or a list: the time complexity for iterating an array or list is $O(n)$ (i.e., the required time increases proportionally with the number of individuals). In the next section we present a more efficient data structure.

III. AN AUGMENTED B-TREE FOR WEIGHTED RANDOM SELECTION OF INDIVIDUALS

A. Data structure

To move an agent-based SIR model forward in time requires summing the individual infection and recovery rates, drawing a time till the next event and selecting the event type and the individual. As the individual rates may differ, the selection of an individual is a weighted random selection. After an individual has been selected, he or she is moved to the next state.

To prevent $O(n)$ time complexity, we would like an alternative to simply iterating over a list with individual rates. The main requirements for an alternative data structure or algorithm are:

- rapid weighted random selection of elements
- quick and easy insertion and removal of elements (or of updating the rates)

A data structure that fulfils these requirements is a balanced search tree with nodes that are augmented [9] with a record of the sum of an attribute of the child nodes. We chose a standard B-tree [10] as the basis. B-trees are widely used in relational databases and have $O(\log(n))$ time complexity for search, delete and insert actions [10]. Fig. 1 illustrates a B-tree

 ISBN: 978-1-61208-308-7

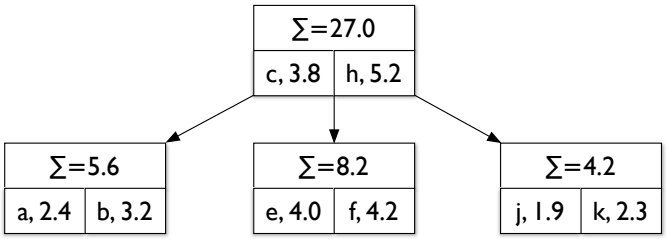


Fig. 1. Augmented B-tree for rapid selection weighted by individual rate. Each node contains *i.* the sum of the values of the elements in that node and in all subtrees of that node, *ii.* an ordered list of key, value pairs (e.g., a, 2.8 for smallest element in the tree), and *iii.* pointers to child nodes with keys intermediary between the keys of the elements left and right of the pointer. In this example, the tree contains contact rates (the values) of individuals identified by single character as key.

(of order 2, i.e., with either 1 or 2 elements per node) in which the nodes have been augmented with the sum of the values of the elements in that node and all subtrees of that node. In this example, each element would represent an individual denoted by a key (single lowercase character) and a value (e.g., contact rate). To select a random individual weighted by contact rate, we proceed as follows. First, a random number x is drawn from a uniform distribution between 0 and the sum of all values in the tree. Next, x is compared with the sum of the leftmost child node, and if the sum of that node is less than x, subtract that sum from x, and continue to the value of the leftmost element in the root node, the sum of middle child node etc. As soon as the current value of x is less than the value of an element or child node, the element or child node will be selected. Suppose the sum of the values is 27.0 (see Fig.1) and we have drawn $x = 10$; as the sum of the left child node $5.6 < 10$, $x \leftarrow 4.4$; as the value of $c, 3.8 < 4.4$, $x \leftarrow 0.6$ and we descend to the middle child node; as the value of e, $4.0 > x$, e is selected.

B. Expected performance

For each level in the tree, at most 4 comparisons (2 values within the node itself and 2 out of 3 subtree sums referenced by pointers) are needed to either find the required element or find the subtree containing the element. Thus, the number of comparisons for weighted random selection increases linear with the number of levels whereas the number of elements in a tree increases exponentially with the number of levels of the tree. Therefore, time complexity of weighted random selection is $O(\log(n))$. Time complexity of insert, delete and update actions in an augmented B-tree is also $O(\log(n))$ as only the sums of the nodes on the path leading to the element need updating for these actions. Selecting by key is the same as in standard B-trees, i.e., $O(\log(n))$.

As for standard B-trees [10], the space complexity for the augmented B-tree is $O(n)$; the pointer structure is identical to that of a B-tree but additional space is required for storing the sums. The space for storing the sums decreases with increasing number of elements per node and can be tuned. Note that the values do not have to be stored in the tree if the values can be referenced through the key.

TABLE I. PERFORMANCE OF A JAVA ARRAYLIST VS. AN AUGMENTED B-TREE FOR WEIGHTED RANDOM SELECTION. TIME IN μSECS PER SELECT (AVERAGE±SEM OF 10 RUNS OF 5,000 SELECTS EACH). ALL DIFFERENCES BETWEEN ARRAYLIST AND AUGMENTED B-TREE WERE SIGNIFICANT AT P < 1E-6 (STUDENT T-TEST).

Number of elements	ArrayList	Augmented B-tree
10,000	6.6±0.1	0.32±0.01
20,000	13.9±0.1	0.50±0.04
50,000	35.0±0.3	0.65±0.01
100,000	71.5±0.5	0.88±0.01
200,000	174±1	1.13±0.01
500,000	522±2	1.40±0.01
1,000,000	1073±3	1.71±0.02

C. Measured performance

Table 1 shows the performance of weighted random selection using a Java ArrayList versus an augmented B-tree. The time required for 5,000 weighted random select actions was determined for increasing numbers of elements in the data structure. Each test was performed 15 times. To allow the Java VM to warm up, only the final 10 runs were used for calculating the average and SEM. All tests were performed on a MacBook Pro with 8 GB RAM and a 2.8 GHz Intel Core i7 processor on OS X 10.8.4 using the Java SE 6 runtime. Minimum and maximum Java heap space was set to 768 MB. A Java software library including the augmented B-tree will be published as open source on www.skardahl.com, the website of Skardahl BV, before October 27, 2013.

The figures in Table 1 show that the augmented B-tree has much better performance than the Java ArrayList, even when just 10,000 elements are present in the data structure. In addition, the table show that the time complexity is about $O(n)$ for the ArrayList (about a 10-fold increase in time from 100,000 to 1,000,000 elements) whereas for the augmented B-tree the increase is about proportional to $\log(n)$. The amount of memory required for the augmented B-tree was about six times that of a Java ArrayList: for 1 million elements 147 MB for the augmented B-tree vs. 25 MB for the ArrayList.

IV. APPLICATION OF THE AUGMENTED B-TREE TO AN INDIVIDUAL-BASED SIR MODEL

Fig. 2 shows the dynamics of an agent-based SIR model with and without heterogeneity in contact rate. Heterogeneity decreases variability and causes an earlier and higher peak in the number of infecteds whereas the fraction susceptible remaining after the epidemic has died out is the same (not shown). The data shown in fig. 2 were generated by 20 simulation runs each modelling a population of size 100,000 with individuals with either the same or different contact rates. The 20 simulation runs took 5 seconds in total. When an ArrayList was used for weighted random selection the 20 runs took 12 minutes in total. Using the augmented B-tree therefore caused a speed-up of a factor 140. For larger population sizes the difference would even be larger.

V. USE OF THE AUGMENTED B-TREE FOR EPIDEMICS IN AGE STRUCTURED POPULATIONS

So far, we focused on random selection of elements weighted by the value of the element. Although the elements

 ISBN: 978-1-61208-308-7

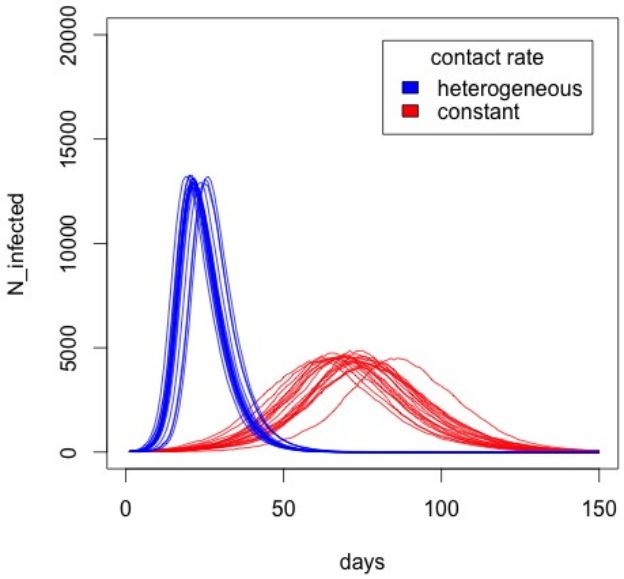


Fig. 2. Dynamics of a SIR model with 100,000 individuals, an average effective contact rate (equal to transmission probability per contact multiplied with contact rate) of 0.35 per day, and a disease duration of 4 days. The parameters are equivalent to R0 = 1.4. The red lines (peaking around 75 days) show the time course of the number of infected individuals for a model where all individuals have the same effective contact rate and the blue lines (that peak around 25 days) show the same for heterogeneous individual contact rates which were drawn from an exponential distribution with mean 0.35 per day.

are ordered by key, the ordering was irrelevant for the simple individual based SIR model.

When modeling SIR dynamics in an age-structured population, the obvious key to use is age (or birthdate). When using age as a key, the augmented tree allows summing the individual effective contact rates of the infecteds by age range. For each age range, the summed rate can be distributed over age ranges of susceptibles according to a contact matrix.

An additional feature of the augmented tree (not related to scheduling transmission events) is that we can get a quick response to queries for the sum of attribute values in a certain key range. For example, in an agent-based SIR model we could find out which age group (or birth cohort) causes most new infections by using birthdate as key and the effective contact rate as attribute value and querying different age ranges of infecteds. In the same way we could find out which age group is most subject to new infections by querying aggregate contact rates in age ranges of susceptibles. As for selection and update, the time complexity of key range queries is $O(\log(n))$.

VI. Discussion

We have described a data structure that can speed up agent-based simulations of infectious disease dynamics in large populations by a factor of 100 or more. The augmented B-tree that we have presented enables more realistic modeling of individual heterogeneity (e.g., in contact rate) while at the same time offering the option to easily aggregate the individual rates by key range thereby providing insight into the groups causing and experiencing the force of infection.

A similar algorithm as presented here has been described for the simulation of chemical reaction kinetics. Gibson and Bruck [11] developed a logarithmic scaling version of the SSA (stochastic simulation algorithm) using an indexed priority queue or binary tree for a more efficient way to select the chemical reaction that will fire next. Slepoy et al. [12] present a constant-time kinetic Monte Carlo algorithm that could in principle also be used for agent-based models of epidemics. However, the composition-rejection algorithm presented in [12] requires sorting the rates in categories to prevent too many rejections, and is not easy to implement. Also, it does not offer the option to aggregate rates by key range.

We believe that our augmented B-tree is useful in all agent-based models where random selection weighted by individual risk (i.e., rate, susceptibility, etcetera) is required. In addition, the augmented B-tree is useful to aggregate individual attribute values (e.g., rates), optionally by key range.

Acknowledgment

The first author would like to thank Marijn Bom for support and inspiration, and Sake de Vlas, Luc Coffeng and Richard Steen for stimulating discussions.

References

[1] K. K. Orroth, E. E. Freeman, R. Bakker, A. Buvé, J. R. Glynn, M.-C. Boily, R. G. White, J. D. F. Habbema, and R. J. Hayes, "Understanding the differences between contrasting hiv epidemics in east and west africa: results from a simulation model of the four cities study," *Sexually transmitted infections*, vol. 83, no. suppl 1, pp. i5–i16, 2007.

[2] J. A. Hontelez, S. J. de Vlas, F. Tanser, R. Bakker, T. Bärnighausen, M.-L. Newell, R. Baltussen, and M. N. Lurie, "The impact of the new who antiretroviral treatment guidelines on hiv epidemic dynamics and cost in south africa," *PloS one*, vol. 6, no. 7, p. e21919, 2011.

[3] N. M. Ferguson, D. A. Cummings, S. Cauchemez, C. Fraser, S. Riley, A. Meeyai, S. Iamsirithaworn, and D. S. Burke, "Strategies for containing an emerging influenza pandemic in southeast asia," *Nature*, vol. 437, no. 7056, pp. 209–214, 2005.

[4] J. M. Epstein, "Modelling to contain pandemics," *Nature*, vol. 460, no. 7256, pp. 687–687, 2009.

[5] W. Kermack and A. McKendrick, "A contribution to the mathematical theory of epidemics," *Proc. R. Soc. Lond. A.*, vol. 115, no. 772, pp. 700–721, 1927.

[6] V. Grimm and S. F. Railsback, *Individual-based modeling and ecology*. Princeton university press, 2005.

[7] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *The journal of physical chemistry*, vol. 81, no. 25, pp. 2340–2361, 1977.

[8] H. W. Hethcote and J. A. Yorke, *Gonorrhea transmission dynamics and control*. Springer Berlin, 1984, vol. 56.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. The MIT press, 2001.

[10] R. Bayer and E. McCreight, *Organization and maintenance of large ordered indexes*. Springer, 2002.

[11] M. A. Gibson and J. Bruck, "Efficient exact stochastic simulation of chemical systems with many species and many channels," *The journal of physical chemistry A*, vol. 104, no. 9, pp. 1876–1889, 2000.

[12] A. Slepoy, A. P. Thompson, and S. J. Plimpton, "A constant-time kinetic monte carlo algorithm for simulation of large biochemical reaction networks," *The journal of chemical physics*, vol. 128, p. 205101, 2008.

 ISBN: 978-1-61208-308-7

Estimating Energy Efficiency of Data-Link Layer in System Level Performance Evaluation

Subayal Khan, Jukka Saastamoinen, Jyrki Huusko and Juha Korpi
VTT Technical Research Centre of Finland
Oulu, Finland
e-mail: {firstname.lastname}@vtt.fi

Jari Nurmi
Department of Computer Systems
Tampere University of Technology
Tampere, Finland
e-mail: jari.nurmi@tut.fi

Abstract—**Modern distributed embedded systems are composed of a number of mobile devices, which have limited battery life. The design of distributed embedded systems is therefore challenging and data-link layer plays an important role in end-user experience by ensuring reliable error free communications. Both the non-functional properties such as end-end delays and frame error rate, and energy consumption of the data-link and upper layer protocols must be thoroughly investigated in order to ensure optimal distributed system design. To achieve this goal, we propose a novel framework to estimate the energy consumption and non-functional properties of the Medium Access Control (MAC) protocols. In this article we elaborate the extensions made for Abstract workload based performance simulation (ABSOLUT) System Level Performance Evaluation (SLPE) approach and the corresponding methodology to estimate energy consumption of IEEE 802.11 family MAC protocol through a case-study in UDP/IP transmission.**

Keywords-Data-link; ABSOLUT; System Level Performance Evaluation; Distributed Systems; Energy Consumption

I. INTRODUCTION

System Level performance evaluation, has been approached in various ways. A detailed state of the art survey is provided by Khan et al. [1] and is therefore not presented here. In many embedded systems domains such as wireless sensor networks, devices are limited by battery and computation capacity. To operate successfully under these constraints, the wireless devices mostly employ highly specialized MAC protocols such as [1] and [2]. For highly accurate System Level Performance Exploration (SLPE), the system designer must be provided with highly accurate models of applications, middleware, transport, MAC and platform hardware components (such as processors, busses and memories). Different MAC protocol models can be used in the performance model in different iterations of the performance exploration. This will help to evaluate their feasibility before the design and deployment of the devices involved in the distributed system.

In lucrative mobile devices market segments, such as mobile phones and tablets the middleware technologies, application design methodologies, multi-core processors, cloud computing and application specific processors have played a key role in maintaining the good customer perception of the strongest brands like Samsung and Apple [3][4]. The reliable and highly accurate energy consumption of MAC protocols in different use-cases is of fundamental importance since most of the applications used by these devices are distributed [1] [2].

After the performance simulation, both the performance (non-functional properties such as end-end delays and frame error rate) [5] and energy consumption of the chosen MAC protocol must be reported simultaneously for a particular iteration during architectural exploration. This helps to evaluate the feasibility of a MAC protocol (non-functional properties, which must be satisfied by MAC protocol) under the energy and power constraints. After, the simulation, the values of non-functional properties (such as end to end delays and frame errors, etc.) are examined to ensure that their values are under the maximum allowable values for the current use-case. In case of energy consumption, the goal is to find the sole contribution of the employed MAC protocol in the energy consumed by the hardware components of the platforms of devices, which constitute the modelled distributed system.

The aim of the research presented in this article is to measure/estimate the energy consumption due to the MAC/transport protocols via highly accurate and functionally correct state machines models. The functionality of the MAC protocol models mimics the behaviour of modelled MAC accurately to provide reliable estimates of end-end frame delays, loss rates and delays. The highly accurate application workload models representing software implementation of MAC protocols are obtained via ABSINTH-2 [6], PAPI [7], or CORRINA [8]. These workload models [9] provide reliable estimates of busy times of the underlying platform entities due to modelled software implementations of corresponding MAC protocols.

The workload models are combined with the correct behavioural state machine models of contention resolution algorithms employed by MAC protocols [5]**.** The application models can be abstracted out in ABSOLUT via traffic generators to model a variety of use cases and helps to abstract out the workload of applications, thus providing the utilization (busy) times of platform components solely due to MAC protocols implementation [5].

Once the energy efficiency of the targeted MAC protocols is evaluated in isolation, the actual application

 ISBN: 978-1-61208-308-7

workload models of applications are substituted instead of traffic generators. This results in the overall system level performance evaluation of complete distributed embedded systems, as described in [10] [11]. This way, the overall performance evaluation of the distributed system is performed. In this article, we only focus on the energy consumption of the MAC protocols. The methodology presented in this article is described via a case study.

The approach is not limited to any particular data-link protocol or application/system domain. The approach allows the system designer to freely choose between the analytical power models in research for different platform components or any/all of average power, minimum and maximum power values of the platform components.

The rest of the paper is organized as follows: Section II focuses on the related research and identifies the main focus areas of the previously conducted research. This helps to elaborate the importance and uniqueness of the research contribution presented in this article. Section III gives a brief overview of the ABSOLUT performance simulation approach. Section IV and Section V collectively describe the methodology and the analytical models used to estimate the energy consumption of the platform components due to the targeted MAC protocol. In Section VI, the approach is experimented via a case study. Finally the conclusions are presented in Section VII.

II. Related Work

The deployment of energy efficient MAC protocols is important in many embedded systems domains such as wireless sensor networks and mobile devices. In wireless devices, the employed MAC protocols must deliver satisfactory performance in terms of for example end-to-end delays and frame loss rate [5] under the energy constraints due to limited battery [1][2].

So far, the related research has been focused on three main areas in the development of new energy efficient MAC protocols [1][2] such as the software and hardware based energy consumption techniques [12], simulation frameworks for comparing energy saving of MAC protocols [13] and analytical models for estimating the energy consumption of specific MAC protocols such as Y-MAC [14]. The tools and methodologies developed as a result of research contributions do not provide a seamless progression between two steps, i.e., between the selection of energy efficient MAC protocols (under the different use cases) and the system-wide performance evaluation. In the first step, the Application workload models are abstracted out by traffic generators to focus solely on the performance of MAC protocols as described by Khan et al. [5]. This results in choosing the most efficient MAC protocol among different alternatives. Once the MAC protocol is selected, the second step proceeds. In the second step, the actual application workload models are employed (and traffic generators are removed) to evaluate the feasibility of the overall system. In order to obtain reliable performance numbers in both steps, the methodology employs the following important techniques/tools:

1. The highly accurate models of contention resolution techniques, as described by Khan et al. [5].
2. Automatic workload Extraction for MAC protocols via ABSINTH [9] or ABSINTH-2 [6] in the cases where a software implementation is available.
3. In the case where software implementations are not available, it can be estimated via transport layer workload models via CORRINA [8].
4. In order to focus on MAC protocols entirely, the workload models of application layer can be abstracted out, as described by Khan et al. [5].
5. The busy times of the platform hardware components such as processors, memories and busses can be obtained solely due to MAC layers and used to estimate energy consumption via the specifications of the hardware components provided by vendors, such as ARM Holdings [15].

The energy estimation of data-link/MAC protocols via ABSOLUT is fundamentally important to many domains of distributed embedded systems such as wireless sensor networks and mobile devices. During the early design phase, the energy consumption of MAC protocol can be estimated in a variety of use-cases, which helps in the selection of the most appropriate MAC protocol out of a number of available alternatives. This results in a more optimal distributed system design.

III. Introduction to ABSOLUT

The abstract workload based performance simulation (ABSOLUT) approach has been extensively applied for the performance simulation of non-distributed and distributed embedded systems. It follows the Y-chart model [1], consisting of application workloads and platform model [9]. After mapping the workloads to the platform, the models are combined for transaction-level performance simulation in SystemC. Based on the simulation results, we can analyse e.g. processor utilization, memory traffic and execution time. The approach enables early performance evaluation, exhibits light modelling effort, allows fast exploration iteration and reuses application and platform models [9].

A. Application Workload Model

The application workload model has a layered architecture as explained by Kreku et al. [9]. The hierarchical structure of the application workload model is shown in Fig. 1.

 ISBN: 978-1-61208-308-7

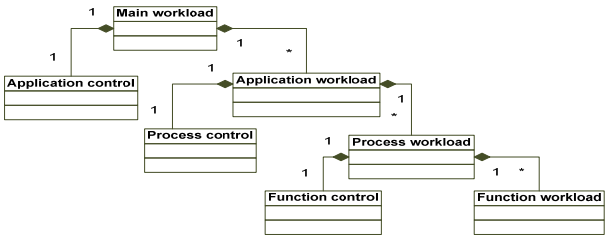


Figure 1. Hierarchical structure of application workload model.

B. Platform Model

The platform model is an abstract hierarchical representation of actual platform architecture. It is composed of three layers: component layer, sub-system layer and platform architecture layer, as shown in Fig 2.

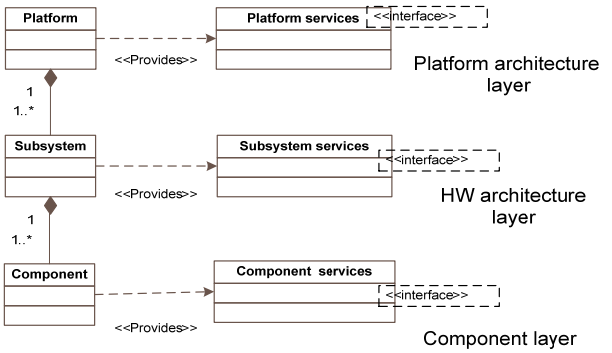


Figure 2. Platform model layers.

Each layer has its own services, which are abstract views of the architecture models. Services in sub-system and platform architecture layers are invoked by application workload models [9].

IV. Estimating Energy Consumption of Data-Link Protocols

The ABSOLUT performance evaluation approach allows the system designer to adjust the parameters for the platform components (prior to simulation) and also reports a variety of performance statistics after simulation. A subset of these parameters and statistics are listed below.

- Adjusting Clock rate of platform components.
- The percentage of platform component consumption (proportion of the total busy times of components) by the different OSI model layers and their protocols.
- Average Packets/sec.
- Average Frames/sec.
- Idle/Busy times of components.
- The number of words written to and read from the memory.
- The collision rate and the maximum and minimum frame delays.
- Maximum and minimum frame delays.
- The busy times of each and every core of a multi-core processor.
- Trace, which validates the functional correctness of the MAC contention resolution algorithms.

The aforementioned parameters and performance statistics are used for measuring the energy consumption of the platform components by different Layers of the Open Systems Interconnection (OSI) model [16]. The total energy consumption of the platform is shown as the sum of energy consumption of all the platform components

$$E_{platform\_total} = E_{comp1} + E_{comp2} + \ldots + E_{compN}, \quad (1)$$

where the E_{compi} is the energy of the i^{th} platform component.

The energy of each platform component is mostly found by multiplying the average power of the component with the busy time. It should be noted that in the case of some components, the power consumed can be given by the analytical formulas as explained afterwards. If the average power consumption of the components is available via vendors, we can write

$$E_{platform\_total} = (Pwr_{comp1})(T_{comp1}) + (Pwr_{comp2})(T_{comp2}) + \ldots + (Pwr_{compN})(T_{compN})$$

$$E_{plaform\_total} = \sum_{n=1}^{N} (Pwr_{compi})\,(T_{compi}). \quad (2)$$

The total busy time of each component (T_{compi}) is due to the sum of busy times (utilization of the component) due to each OSI model layer. ABSOLUT provides the ability to separately report the consumption of data-link and transport layers by abstracting out the application workload models by traffic generators. Therefore the energy consumption of the data-link layer can be given by.

$$E_{plaform\_dl} = \sum_{n=1}^{N} (Pwr_{compi})\,(T_{compi\ DL}). \quad (3)$$

This equation requires the busy time and power consumption of each component in the platform. We now describe the methods for estimating the busy times and power consumption of platform components.

A. Estimating Busy Times of Platform Components

As shown by (3), in order to compute the busy times of the components, the workloads of the OSI model layers considered in the use case must be modelled. Just like the application layer ABSOLUT workload models, the workload models of the data-link layer are also composed of abstract instructions as described by Khan et al. [5] and are executed by the processor models (multi-core or single core) of the ABSOLUT platform models. After execution, the amount of time taken by the platform components to execute these instructions is automatically reported. During the simulation whenever the MAC frame of a transport layer packet is served by the functionally correct data-link layer

 ISBN: 978-1-61208-308-7

MAC protocols, the workload (consisting of abstract instructions) is send to the underlying processing elements of the ABSOLUT platform by the ABSOLUT operating system (OS) model as described in [5].

The data-link layer protocols (for example MAC protocols) and transport layer protocols are modelled as operating system (OS) services in ABSOLUT as explained in [5] The transport layer OS_Service [5] divides the application layer message into individual frames and sends it one by one to the data-link layer OS_Service for transmission over the channel. The higher layer ABSOLUT OS_Service uses the lower layer ABSOLUT OS_Service as described in [5] just as in case of real OSI model layers. The functional correctness of the ABSOLUT OS_Service provides reliable estimates of non-functional properties such as end-end packet and frame delays and loss rates, which play an important role in the end-user experience in a variety of distributed multimedia streaming applications [10]. In other words, in case of data-link layer, whenever an ABSOLUT channel model is sensed idle by the MAC protocol model, the frame is transmitted and the non-functional workload mimicking frame processing in actual platform is send to the ABSOLUT platform model.

The workload per frame transmission can be automatically extracted via ABSINTH-2 tool if the source code of the MAC protocol implementation is available. In cases where the source code of data-link protocols is not available, the workload can be estimated using CORRINA [8]. In this case, firstly the overall workload of message transmission at the transport layer i.e., the workload of TCP/IP BSD API functions is extracted first. Afterwards, this information is used to estimate the workload of a single frame transmission (data-link protocol). The obtained ABSOLUT workload models are non-functional but the handling of the frame (contention resolution and retransmissions etc.) is done in the functionally correct manner, which is confirmed by the trace generated by the corresponding probes [5].

The workload models of ABSOLUT can be estimated via run-time performance statistics based methods called CORRINA if the source code of the data-link protocol is not available. In such cases, firstly the overall workload models at the transport layer (sum of workload due to data-link and transport) is estimated and afterwards the workload of transport layer is subtracted from it to obtain the workload due to data-link layer. The workload due to the processing at the transport layer (excluding underlying MAC layer) is estimated by using the case studies conducted in research as shown in [5]. If the source code of the data-link protocol is available, ABSINTH-2, which is a compiler based technique, can be used to generate highly accurate workload models.

1) CORRINA: First of all, the TCP/IP or UDP socket API functions, which are meant for sending and receiving the messages in distributed applications are identified. In case of TCP/IP, the socket API functions are send() and receive(). Afterwards, a test bench consisting of a client and server is programmed. The messages exchanged between the client and server can have different lengths (sizes in bytes). The data size is limited to a few bytes bytes so that one MAC frame is transmitted by the MAC layer for each call to send() and receive() transport functions at the transport layer. TCP/IP API Functions tagged by CORRINA [8] and the ABSOLUT workloads models are extracted after exchanging a number of messages between the client and server. In this way the average workload of a TCP/IP send() and receive() API function corresponds to the workload per packet transmission.

The extracted workload models can be used as such and mimic the workload per frame transmission. The energy consumption values obtained as a result will always be pessimistic since they also contain the transport layer processing workload. If required, these workload models can be refined by subtracting the overheads of transport layer from the overall workloads obtained (for TCP/IP API functions send() and receive()). The information about the processing overheads of the transport layer is elaborated in [8]. The obtained workload models are mapped to the corresponding ABSOLUT MAC OS_Service and on each frame transmission, the workload/frame is executed by the processor models in the ABSOLUT platform models.

2) ABSINTH-2: This method is especially useful for the workload modelling of MAC protocols which are developed for specialized applications such as WSNs (Wireless Sensor Networks). The implementation of these protocols is usually done in "C" programming language. WSNs have more specific requirements, which include a local unicast or broadcast. The traffic flow is usually from many nodes towards one or a few sinks (most traffic is thus directed in one direction). The individual nodes have periodic or rare communication and must consider energy consumption as a major factor.

An effective MAC protocol for WSNs must have reduced power consumption, shall avoid collisions, should be implemented with a small code size and memory requirements, be efficient for a single application and be tolerant to changing radio frequency and networking conditions [17]. That is why many WSNs employ highly efficient MAC protocols for the transfer of frames over the wireless channels for example NANO MAC [2] and BMAC [18].

After the design and development of these novel MAC protocols, the source code of these protocols can be compiled with ABSINTH patched gcc compiler and executed. After executing the use-case, the profiling information, which is used to generate highly accurate workload models, is obtained. These workload models can be mapped to the ABSOLUT MAC OS_Service to obtain the energy consumption of these protocols on different platforms.

 ISBN: 978-1-61208-308-7

B. Estimating Power Consumption of Platform Components

The power consumption of the individual components (busses, processors and memories etc.) are obtained either via vendor specifications or analytical models. If the average power consumption of a platform component is provided by the vendor, the energy consumption is easily estimated by multiplying the power consumption value with the busy time of the component reported by ABSOLUT performance simulation. For example, if the ABSOLUT platform model contains models of ARM_Cortex A-9 processors, the energy consumption of the processor is simply a product of the busy times obtained by the ABSOLUT performance simulation and the power values available on the following website [15].

If the power values are not provided by the component manufacturer, the analytical models available from the literature are utilized. For example, the power consumption of an un-buffered DDR2 SDRAM (in idle state) is given by [19]:

$$P_{RAM\_idle} = \sum_{i=1}^{n} (S_i * P), \quad (4)$$

where *n* is the number of installed memory modules and *s* is the size (in GB) of the memory modules. The values of "p" for different vendors are mentioned in Table I.

TABLE I: VALUES OF P FOR UN-BUFFERED DDR2 SDRAMS MANUFACTURED BY DIFFERENT VENDORS

Vendor	Value
Kingston	*f/1000*
Samsung	*0.95* f/1000*
Hynix	*1.9* f/1000*
Generic	*1.45* f/1000*

The model of the dynamic power consumption has been derived in [19] by using the RAMSpeed benchmark [20] and is given by the following equation:

$$P_{RAM} = P_{RAM\_idle} + \beta, \quad (5)$$

where β = 7.347 as described in [19]. The equations listed in Table I require the value of frequency (*f*) of the SRDAM in Mega Hertz, which is obtained by accessing its value from the list of ABSOLUT platform component parameters values set by the system designer. Also, the different analytical models for the measurement of power consumption of the network interface (NIC) card are provided in [21]. We consider the linear model given by the following equation:

$$P_{NIC} = P_{idle} + (P_{max} - P_{idle}) * pps. \quad (6)$$

In (6), P_{NIC}, P_{idle} and P_{max} denote the total, idle and maximum power of the network interface card whereas *pps* denotes packets per second, which is reported at the end of ABSOLUT simulation. In many cases, the values of P_{idle} and P_{max} can be obtained from vendor specifications . In cases where the values of P_{idle} and P_{max} are not provided by the vendors, they can be estimated by using the trace based method described by Ebert et al. [22]. The system designer might select a range of values for P_{idle} and P_{max} to determine the range (of allowable P_{idle} and P_{max}) of allowable values feasible for the use case. After the simulation the NIC cards, which fulfil the power and energy budgets can be integrated into the devices.

V. METHODOLOGY

The energy consumption of data-link protocols via ABSOLUT involves the following steps.

1. In the first step, the ABSOLUT workload models for data-link protocols are automatically generated by using ABSINTH-2 or CORRINA. These workload models are assigned to the ABSOLUT MAC OS_Service [5]. Therefore, whenever a MAC frame is transmitted, the estimated processing load is executed on the platform model.
2. After workload extraction of data-link layer, the application workload models are abstracted out by using traffic generators. This is important since we need only the workload of data-link protocols to load the ABSOLUT platform model. In this way, we get the platform utilization of the platform by data-link layer only [5] in isolation.
3. After simulation, the busy time of each platform component is automatically obtained and can be used directly in (3). The busy time of each platform component is then multiplied by the average power of the corresponding platform component to find the energy consumption of the component due to processing load of data-link layer. The power of different platform components can be found as described in this section.
4. In the next step, the use-case is modeled. The system designer initializes traffic generators [5] with desired parameters and employing the model of the simulated MAC protocol.
5. Once the ABSOLUT performance model is finalized by the system designer, the performance model is executed.
6. After simulation, the busy times of all the hardware components in the platform are reported automatically in a text file. The busy times of each platform component are used in (3) for estimating the energy consumption of the platform by the data-link layer.
7. The busy time of each platform component is then multiplied by the average power of the corresponding platform component to find the energy consumption of

 ISBN: 978-1-61208-308-7

the component due to processing load of data-link layer. The power of different platform components can be estimated in a variety of ways.

8. The sum of energy consumption of all platform components amounts to the overall energy consumption of the platform as described in (3).

The aforementioned methodology for estimating energy consumption of data-link protocols is summarised in Fig. 3.

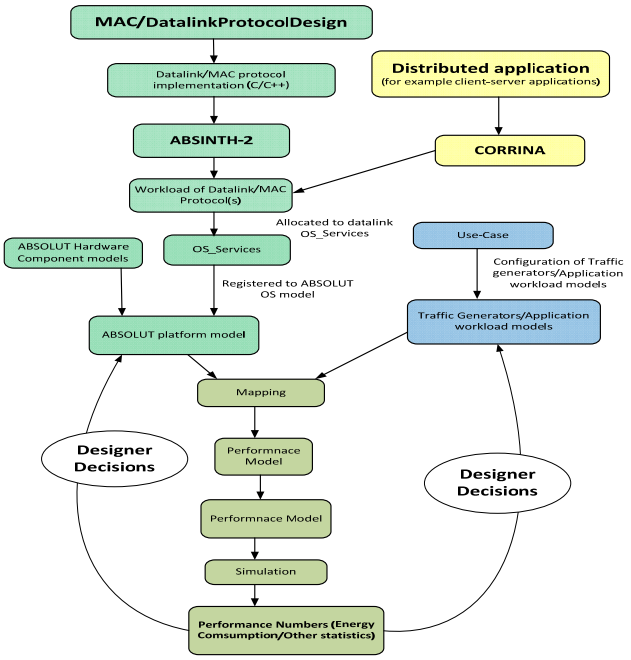


Figure 3. Measuring energy consumption of data-link layer via ABSOLUT

The methodology is not restricted to the energy consumption of data-link layer and can be used to estimate the energy consumption of other layers of the OSI model.

VI. CASE STUDY "IEEE WLAN (IEEE 802.11 DCF)"

After elaborating the methodology, we now describe two case studies, which validate the approach. In both the case studies, CORRINA was used to estimate the workload of the data-link layer as described in previous sections. The UDP/IP transport protocol was used in the test bench consisting of a client server application. The client and servers merely exchange messages of fixed lengths (in Bytes). The message size was limited to a few bytes to ensure the transmission of a single frame for each exchanged message. In this way, the workload per frame transmission is estimated. Therefore, whenever an ABSOLUT OS_Service transmits a frame, the corresponding workload is send to the underlying platform processors by ABSOLUT MAC model for processing. For pessimistic results, we included the overheads of transport layer, though more accurate (and less pessimistic) results can be obtained by studying the research articles focusing on the overheads of TCP/IP or UDP protocols [8]. The workloads used in both the case studies are shown in Table II.

TABLE II: PROCESSING TIMES OF TCP/IP API FUNCTIONS AND ABSTRACT INSTRUCTIONS OF CORRESPONDING ABSOLUT WORKLOAD MODELS. WORKLOADS OF ONLY SEND() AND RECEIVE() API FUNCTIONS ACT AS ESTIMATES OF ABSOLUT PER-FRAME TRANSMISSION WORKLOADS. THESE WORKLOADS ARE USED IN THE ABSOLUT DATA-LINK LAYER MODELS. CORRINA [8] WAS USED FOR AUTOMATIC WORKLOAD MODELING.

NoTA API funcitons	**Average Execution Times on Intel Core i5 Processor based platform. Message size=2 Bytes**	**ABSOLUT workload models (abstract instructions).Workload models of send() and receive functions represent the workload/frame**

socket	3196 usec	m_host->execute(58274432);
		m_host>read(Address(0),45582561,32);
		m_host->write(Address(0),0,32);

bind	446 usec	m_host->execute(1296);
		m_host->read(Address(0),3375,32);
		m_host->write(Address(0),0,32);

listen	374 usec	m_host->execute(1168);
		m_host->read(Address(0),3300,32);
		m_host->write(Address(0),228,32);

accept	592 usec	m_host->execute(372284);
		m_host->read(Address(0),4491,32);
		m_host->write(Address(0),772,32);

receive	442 usec	m_host->execute(283971);
		m_host->read(Address(0),85133,32);
		m_host->write(Address(0)23149,32);

send	429 usec	m_host->execute(67884);
		m_host->read(Address(0),391062 ,32);
		m_host->write(Address(0), 52653,32);

From this point onwards, we only describe the simulation scenarios and the results.

A. Overview of WLAN (IEEE 802.11 DCF)

In a wireless network where a number of stations contend for the wireless medium, if multiple stations sense the channel busy and defer their access, they will also virtually simultaneously find that the channel is released and then try to seize the channel. As a result, collisions may occur. In order to avoid such collisions, IEEE 802.11 Distributed Coordination Function (DCF) is employed, which requires a station wanting to transmit, to first listen to the channel to check its status (occupied or not) for a DCF Inter-frame Space (DIFS) interval. The IEEE 802.11 DCF [5] can be shown in the form of a flow chart, as shown in Fig. 4.

 ISBN: 978-1-61208-308-7

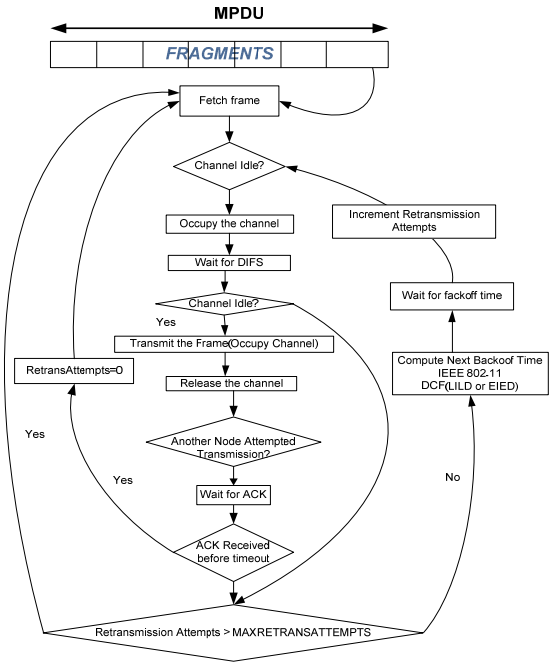


Figure 4: Flow chart of IEEE 802.11 DCF

In ABSOLUT, the MAC protocols, as well as transport layer protocols [5], such as TCP and UDP, are also modelled as services by deriving them from the same base class OS_Service, which provides the scheduling and synchronization mechanism. The transport layer services then request the Layer-2 services such as IEEE 802.11 DCF for transmission of individual frames of a transport layer packet. The do_service() method of the OS_Service base class is implemented by the derived class to provide the functionality of IEEE 802.11 DCF, as explained by Khan et al. [5]. The do_service() method spams a separate frame transmission function for handling each request of frame transmission from the transport as described in [5]. The simulation parameters used for the conducted case study are mentioned in Table III.

TABLE III: EXPERIMENT PARAMETERS (IEEE 802.11B)

Parameters	Values
SIFS	10 µs
DIFS	50 µs
Slot Interval(SLOT)	20 µs
Preamble Length	144 bits
PLCP header Length	48 bits
Channel bit rate	2 Mbps
CW_{min}	31
CW_{max}	1023
CW_o	32
EW	16

Whenever a frame transmission is serviced by ABSOLUT MAC protocol model, the workload per frame transmission estimated by CORRINA is executed by the ABSOLUT platform model. After simulation, the busy times and other performance statistics (for example, busy times of components and packets/sec, etc.) are obtained and used to automatically estimate the energy utilization of modelled platform.

B. Simulation Scenario

The simulations are carried out in WLAN environment and consist of 11 nodes, i.e., 10 wireless nodes sending data to an access point. Each node transmits 100 packets to the access point. The transmission of packets occurs after intervals obtained by configuring a Constant Bit Rate (CBR) Traffic Generator. The exact configuration of the traffic generator is shown in Fig. 5. All the network nodes are within the transmission range (form a collision domain) of each other. The access point acts as the only destination for the clients. 802.11 has a variety of standards, the standard simulated in the case study was 802.11b. If the channel is sensed busy, the nodes enter the collision avoidance phase in which each node executes the exponential back-off algorithm [23], as shown in Fig. 4. The nodes wait for a random time interval distributed uniformly between [0,CW] × SLOT. The Contention Window (CW) values can vary between CW_{min}(31) and CW_{max} (1023). The slot value (SLOT) for 802.11b is 20µs. The aforementioned parameters for 802.11b are listed in Table III. Typically, 802.11b products degrade the bit rate from 11Mbps to 5.5,2 or 1 Mbps [23]. We use the bit rate of 2Mbps during the simulation. The bit rate can be changed dynamically during simulation if desired with minor modelling effort. The CBR traffic generator available in ABSOLUT simulation framework is derived by the ns-2 CBR traffic generator [5]. The traffic generators can be configured with different bit rates and packet sizes via simple interface functions, as shown in Fig. 5.

```
//Decide the simulation parameters
double AverageBurstTime=.0025;      //In seconds, means 2.5 milliseconds
double InterFrameTime  = 1   ;      //Means 1 seconds
double SinglePktSize   = 512 ;      //Sinlge packet lent
double Bitrate         =2000000;    //2 Mega bits per second


//Make three generators.
SSConfig::Instance()->SetTrafficGenerator(CBR_TRAFFIC_GENERATOR,
AverageBurstTime,InterFrameTime, Bitrate,SinglePktSize);
```

Figure 5: An example configuration of the CBR traffic generator. Packet Length = 512 Bytes. Data rate = 2 Mbps. Average Burst Time = 0.0025 seconds. Inter Frame Space = 1 second.

 ISBN: 978-1-61208-308-7

C. Platform Model

Each ABSOLUT platform model used in the case study (all nodes and access point) consists of an ARM Cortex-A9 multi-core processor [10] model including four processing cores along with SDRAM, a POWERVR SGX40 graphics accelerator and an Image signal processor. This is shown in Fig. 6. These component models are connected via an AMBA bus model. The processor cores are clocked at 1 GHz and the DDR2 RAM operates at 800 MHz. The Bus model operates at 800MHz. In ABSOLUT methodology, the application models contain approximate timing information. Thus the execution platform is modelled at transaction level following OSCI TLM2.0 standard [9].

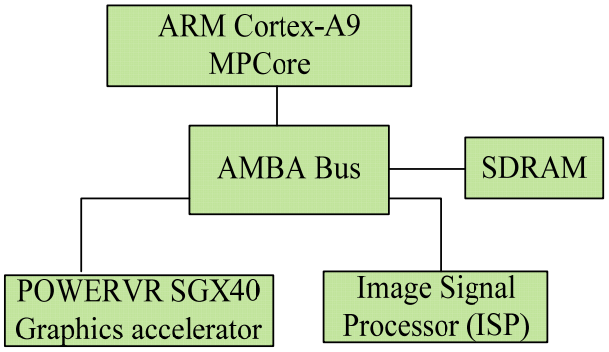


Figure 6: ABSOLUT platform model

Each processor core in the processor model has an L1 instruction and L1 data cache as shown in Fig. 7.

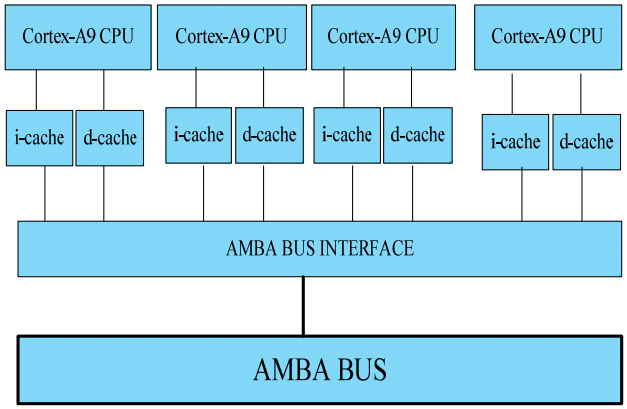


Figure 7: Diagram showing the quad-core processor (ARM Cortex A9 multi-core processor) models used in the performance simulation

D. Simulation Results

The energy and average power consumption of the platforms hosting the access point and the network nodes are shown in the following Table IV.

TABLE IV: ENERGY CONSUMPTION OF ACCESS POINT AND AVERAGE ENERGY CONSUMPTION OF THE WIRELESS NODES DUE TO IEEE 802.11B.

Platform Component	Access Point	Wireless Nodes(Average)
ARM-CORTEX-A9 MPCore	1341.35 kJ	711.694 kJ
NIC (Network Interface Card)	85.6 J	86.2 J
ISP	0 J	0 J
POWERVR SGX40	0 J	0 J
SDRAM	101.013 J	93.064 J
BUS	0.012984 J	0.011809 J
Total Energy of Platform	***1341.5366 kJ***	***711.8732 kJ***

Table V and Table VI present the transport layer and data link layer statistics, which are were collected with the ABSOLUT simulation framework.

TABLE V: UDP PROTOCOL STATISTICS

Average Transport Delay	284.51 ms
Maximum Transport Delay	2456.05 ms
Minimum Transport Delay	0 ms

TABLE VI: 802.11B MAC STATISTICS

Average CW Size	849.008
Maximum CW Size	2048
Average Backoff Time	34.908 ms
Maximum Backoff Time	102.4 ms
Minimum Backoff Time	0 ms
Average MAC Transmission Delay	54.2947 ms
Maximum MAC Transmission Delay	330.33 ms
Minimum MAC Transmission Delay	0 ms
Average Retransmission Attempts per Transmission	2.39828
Minimum Retransmission Attempts per Transmission	0
Maximum Retransmission Attempts per Transmission	7
Total Transmissions	25623
Total Collisions	18083
Collision Probability	0.705733
Average Collisions Per 100 Sec	2804

In our simulation scenario, the detailed statics where collected from UDP protocol and IEEE 802.11b medium access control layer.

VII. CONCLUSION AND FUTUREWORK

The data-link and transport layer models in ABSOLUT offer enormous flexibility and employ separation of concerns, which helps the system designer to modify the models according to modeling objective.

For example, in order to implement a new contention resolution scheme, the system designer only needs to implement or modify the state machine of the algorithm [5]. The energy consumption as well as performance of a

 ISBN: 978-1-61208-308-7

particular OSI model layer can be obtained in isolation by probes and abstracting the workload models of other layers by delays. The performance statistics of the platform components can also be obtained at various levels of detail, for example in [10], the authors have demonstrated the performance of individual cores of the processor models. Also, the other component specific performance statistics such as cached hits/misses for cache models can also be obtained. Due to the availability of different types of traffic generators in ABSOLUT, various scenarios can be simulated. The ABSOLUT component library [9] allows the system designer to model a variety of widely used platforms [10]. In this work, the performance (in terms of energy consumption) of a widely used data-link protocol was demonstrated via ABSOLUT methodology. The same methodology can be used for the performance and energy evaluation of other data-link protocols. In future, the methodology can be further enhanced by providing a library of state machines ABSOLUT models for widely used data-link protocols. This will further shorten the time needed for selecting the appropriate data-link protocol.

ACKNOWLEDGEMENTS

This work was performed in the EU-ECONET project funded by the European Union.

REFERENCES

[1] S. Khan, E. Ovaska, K. Tiensyrjä, and J. Nurmo Nurmi, ”From Y-chart to seamless integration of application design and performance simulation,” Proc. International Symposium on System-on-Chip - SOC. Tampere, Finland, 29-30 Sept. 2010. IEEE. Piscataway, NJ, USA, 2010, pp. 18-25

[2] J. Haapola, “NanoMAC: A Distributed MAC Protocol for Wireless Ad Hoc Sensor Networks,” Proc. XXVIII Convention on Radio Science & IV Finnish Wireless Communication Workshop, 2003, pp. 17-20.

[3] http://www.samsung.com/fi/#latest-home, [retrieved: August, 2013].

[4] http://www.apple.com, [retrieved: August, 2013].

[5] S. Khan, J. Saastamoinen, M. Majanen, J. Huusko and J. Nurmi, “Analyzing Transport and MAC Layer in System-Level Performance Simulation,” Proc. International Symposium on System-on-Chip 2011, Tampere, Finland, October 31 - November 2, 2011.

[6] J. Saastamoinen and J. Kreku, ”Application workload model generation methodologies for system-level design exploration,” Proc. IEEE Conference on Design and Architectures for Signal and Image Processing, DASIP 2011. Tampere, Finland, 2-4 Nov. 2011, pp. 254-260

[7] http://icl.cs.utk.edu/papi/, [retrieved: August, 2013].

[8] S. Khan, J. Saastamoinen, J. Huusko, J.-P. Soininen and J. Nurmi, “Application Workload modelling via Run-Time Performance Statistics,” IJERTCS 2013. Int. J. Embedded and Real-Time Communication Systems *(Accepted). To appear.*

[9] J. Kreku et al., “Combining UML2 Application and SystemC Platform Modelling for Performance Evaluation of Real-Time Embedded Systems,” Hindawi Publishing Corporation. EURASIP Journal on Embedded Systems. Volume 2008, Article ID 712329, 18 pages, doi:10.1155/2008/712329.

[10] S, Khan, J. Saastamoinen and J. Nurmi, “System-Level Performance Evaluation of distributed multi-core NoTA systems,” Proc. 2nd IEEE International Conference on Networked Embedded Systems for Enterprise Applications. NESEA 2011, Fremantle, Perth, Australia. 8th - 9th December 2011.

[11] S. Khan, J. Saastamoinen, K. Tiensyrjä and J. Nurmi, “System Level Performance Simulation of distributed GENESYS Applications on multi-core platforms.” Proc. 9th IEEE Int. Conf. on Dependable, Autonomic and Secure Computing (DASC), 2011.

[12] P. Hurni, B. Nyffenegger, T. Braun and A. Hergenroeder, “On the accuracy of software-based energy estimation techniques,” in Wireless Sensor Networks (pp. 49-64). Springer Berlin Heidelberg, 2011, pp. 49-64.

[13] G. P. Halkes, T. van Dam and K. G. Langendoen, “Comparing energy-saving MAC protocols for wireless sensor networks,” in Mobile Networks and Applications, 10(5), 2005. pp. 783-791.

[14] V. Ramchand and D. K. Lobiyal, “An analytical model for Energy Consumption in Y-MAC Protocol,” International Journal of Computer Science, 9, 2012.

[15] http://www.arm.com/products/processors/cortex-a/cortex-a9.php, [retrieved: August, 2013].

[16] H. Zimmermann, “OSI Reference Model – The ISO Model of Architecture for Open Systems Interconnection,” IEEE Transactions on Communications, vol. COM-28, no. 4, April 1980, pp. 425-432.

[17] I. Lee, J. Y.-T. Leung and S. H. Son (ed.), “Handbook of Real-Time and Embedded Systems,” Chapman and Hall/CRC (July 23, 2007) p. 800, ISBN-10: 1584886781, ISBN-13: 978-1584886785.

[18] J. Polastre, J. Hill, and D. Culler, “Versatile Low Power Media Access for Wireless Sensor Networks,” Proc. ACM SenSys, November 2004.

[19] O. Mämmelä, M. Majanen, R. Basmadjian, H. De Meer, A. Giesler and W. Homberg, “Energy-aware job scheduler for high-performance computing,” Computer science – Research and Development, vol. 27, issue 4, November, 2012, pp. 265-275, DOI: 10.1007/s00450-011-0189-6

[20] http://alasir.com/software/ramspeed/, [retrieved: August, 2013].

[21] J. C. Cardona Restrepo, C. Gruber and C. M. Machuca, “Energy Profile Aware Routing,” Proc. First International Workshop on Green Communications, IEEE International Conference on Communications (ICC), June 2009. Dresden, Germany.

[22] J.-P. Ebert, B. Burns, and A. Wolisz, “A trace-based approach for determining the energy consumption of a wlan network interface,” Proc. European Wireless, Florence, Italy, February 2002, pp. 230-236.

[23] G. Berger-Sabbatel, F. Rousseau, M. Heusse and A. Duda, “Performance anomaly of 802.11b,” Proc. The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003), San Franciso, CA, USA, March 30 - April 3, 2003.

 ISBN: 978-1-61208-308-7

Modeling Planned and Unplanned Store Stops for the Scenario Based Simulation of Pedestrian Activity in City Centers

Jan Dijkstra and Joran Jessurun
Department of the Built Environment
Eindhoven University of Technology
Eindhoven, The Netherlands
e-mail: {j.dijkstra,a.j.jessurun}@tue.nl

Abstract—**Micro-scale agent-based modeling can be used for the simulation of pedestrian movement for low and high density scenarios and of the effect of changes in an environment. Such models can also be used for pedestrian dynamics in city centers to show the design effects in the shopping environment. The main contribution of this paper is to introduce the implication of planned and unplanned store visits within a simulation framework for pedestrian movement simulation. This paper reports findings of planned and unplanned store visits by using Monte Carlo simulation.**

Keywords- Monte Carlo Simulation, Activity Agenda, Agent Based Modeling, Pedestrian Dynamics

I. INTRODUCTION

Agent-based modeling is a computational methodology that allows us to create, analyze, and experiment with artificial worlds populated by agents. A specific research area is micro-scale agent-based modeling that can be used for the simulation of pedestrian movement for low and high density scenarios and for the effect of changes in the environment. Such models can also be used for pedestrian dynamics in city centers to show the design effects in the shopping environment. In this context, Ali and Moulin [1] describe their multi-agent simulation prototype of customers' shopping behavior in a mall. Therefore, a multi-agent model to simulate pedestrian dynamic destination, route and scheduling behavior is under development, where the simulation of movement patterns is embedded in a more comprehensive model of activity travel behavior.

Representation is a main issue in simulating pedestrian dynamics. One can distinguish the representation of the pedestrian environment and the representation of pedestrians. In the domain of a city center, representation of a pedestrian environment includes the geometry of the shopping environment such as stores and streets, the network as a cellular grid, and pedestrian objects. Pedestrian representation includes socioeconomic characteristics, speed, goals, familiarity with the environment, and activity agenda. It is assumed that pedestrians perceive their environment and that they are supposed to carry out a set of activities. For completing an activity, pedestrians spend time in stores. As a consequence, time duration influences their movement behavior over the network.

Although a 3D presentation of pedestrians and the pedestrian environment for the simulation of pedestrian movement is the ultimate goal, it is nevertheless meaningful to test the underlying principles in an appropriate 2D representation of pedestrians and their environment. NetLogo can be used as a simulation toolkit because it is a suitable simulation framework that supports modeling, simulation and experimentation. It also offers skeletons of agents and their environment, and interoperability (e.g., Geographic Information System (GIS)). We will use shape-file information of the environment and network structure for visualizing the 2D environment and NetLogo for the actual simulation.

The main subject of this paper is to introduce the implication of planned and unplanned store visits within a simulation framework for pedestrian movement simulation. This framework involves an agent-based model that provides an activity agenda for pedestrian agents that guides their shopping behavior in terms of destination and time spent in shopping areas. In order to implement the activity agenda, pedestrian agents need to successively visit a set of stores and move over the network. It is assumed that pedestrian agents' behavior is driven by a series of decision heuristics. Agents need to decide which stores to choose, in what order and which route to take, subject to time and institutional constraints. It is assumed that pedestrian agents are in different motivational states. They may at every point during the trip have general interests in conducting particular activities, without having decided on the specific store to visit, but they may also be in a more goal-directed motivational state in which case they have already decided which store to visit. The motivational states are of influence on the impulse and non-impulse store choice processes and therefore on the planned and unplanned visits to a store. All these aspects affect pedestrian agents' time duration in visiting stores. Pedestrian agents move over a street network and are part of a pedestrian flow in this street network. However, pedestrian agents can be temporarily removed from the pedestrian flow by visiting a store and participating again in the pedestrian flow after visiting that store. In that case, the time spent by a pedestrian agent in a store is relevant. For the simulation run this time duration is determined by a Monte Carlo simulation [2]. In this paper, the focus is on the number of planned and unplanned store visits because that determines the activity agenda. Successful completing an activity by a store visit influences the remaining planned and unplanned store visits. The findings from the collected data of the number of planned and unplanned store visits indicate that this number meets the Gamma distribution, and that this number also depends of

 ISBN: 978-1-61208-308-7

the motivation and some socio-economic characteristics of the visitor to the city center.

This paper discusses successively pedestrian movement simulation in section II, simulation process in section III, and planned and unplanned store stops in section IV. A discussion about the conclusions and future directions will conclude this paper in section V.

II. PEDESTRIAN MOVEMENT SIMULATION

Pedestrian movement simulation consists of pedestrian agents and a simulation environment, which consists of a street network and a set of stores. Polygons are used to indicate borders and functional areas such as walkways. Each cell in the network has information about which agents and polygons occupy it. Also, it contains information about other features such as appearance of stores that are observable from that cell. A pedestrian agent moves with his own behavior and personal characteristics. Every time step, there is an update about agent's positions. The cellular network provides percepts to the pedestrian agent and the pedestrian agent performs actions based on their percepts. Behavioral principles drive the pedestrian movement. Details of the equations representing these behavioral principles are beyond the scope of this paper and are presented in Dijkstra et al. [3], but are related to the *Agent Loop* in Fig. 1. They include the perceiving of the environment, the possible match of the percepts with the activity agenda and as a consequence the determination of the activation to a store with as a result a completed activity with a consequence for the activity agenda.

In the case of the test ground of the city center, each store consists of a cell containing store information. A street network consists of cells with cell information. Pedestrian agents are situated in the cells of the network, namely a street cell or a store cell. The network is irregular because a clear border between a store and adjacent cell is desired. Additionally, each cell in this network is identified by its node and these nodes are linked together.

For populating pedestrian agents in the environment and for attaching activity agendas to pedestrian agents, a Monte Carlo simulation is used which implies that the behavior of each pedestrian agent is simulated by a series of draws of random numbers from successive probability distributions [4]. These probability distributions are based on real data collections, such as time spent in a store, attaching inner lane or outer lane as an entry point, speed, and pedestrian characteristics (gender, age, etc.).

NetLogo is used for the simulation because it easily allows the empirical testing of the principles of the simulation approach. An attractive feature is its ability to integrate GIS data directly into the simulation. With the integration of GIS, pedestrian agents can move around 'real space'. In the simulation, MapInfo data will be integrated with NetLogo. On the basis of this GIS, a network structure with nodes and links will be generated. The nodes are related to polygons in the original drawing, and the links will be determined by the topology of the polygons. The information of the polygons is available for pedestrian agents moving in the network. This information could be store-related information, but also information about the area and perimeter of the polygon. Fig. 1 shows the activity diagram of the simulation setup. The simulation process starts with loading the environment involving GIS information and databases for instance activity agendas and personal characteristics for creating pedestrian agents. The creation of an initial situation at time t_b (beginning time) means that the environment will be populated with pedestrian agents. The simulation run starts at time t_b. The simulation time step includes the creation of zero or more pedestrian agents using the Monte Carlo method: a pedestrian agent would need to be assigned an initial scenario. Also, there is an update of pedestrian agent scenario's that results in pedestrian agent actions and a schedule of the next step.

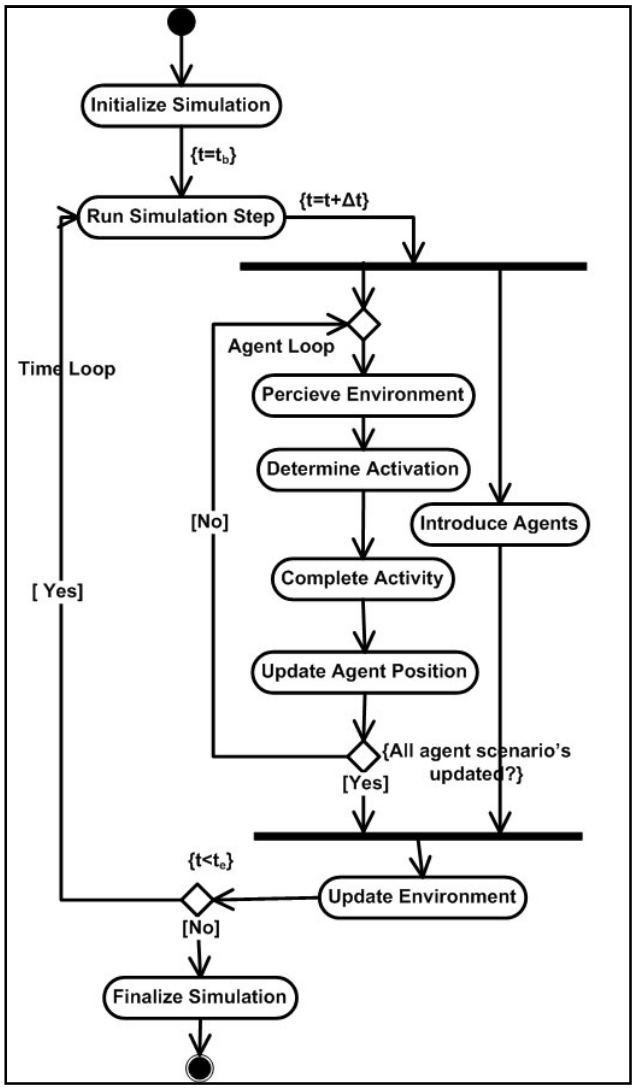


Figure 1. Activity diagram of the simulation setup.

The consequence is the movement to a new position. Then, an update of the environment will be realized. The simulation run stops at time t_e (ending time).

 ISBN: 978-1-61208-308-7

III. SIMULATION PROCESS

This section provides some understanding in the engineering basis of the simulation process using NetLogo [5]. The model structure is based on contexts and projections [6]. The core data structure is called a context that represents from a modeling perspective an abstract population; the objects in these populations are referred as agents. The context provides the basic infrastructure to define a population and the interactions of that population; it creates an abstract environment in which agents exist at a given point in the simulation. The context also holds its own internal state for maintaining the collection of agents. This state can consists of multiple types of data. These provide agents with information about the world in which they interact. In addition, data fields can be maintained by the context. A data field is an n-dimensional field of values with which the agents in a context can interact. These data fields can be directly associated with a physical space. The field is generic, which means each value is derived from a set of coordinates.

Projections take the population as defined in a context and impose a new structure on it. This structure defines and imposes relationships on the population by using semantics defined in the projection; therefore an agent population is realized once a projection is applied to it. This means that projections are added to a context to allow agents to interact with each other. Each context can have an arbitrary number of projections associated with it (1-n relationship); in our case it concerns about two projections.

A feature of NetLogo is the ability to integrate GIS data directly to the simulation; it provides a set of classes that allow shape-files to be displayed. For example, shape-files can be provided by GIS software packages like MapInfo and ArcGIS. A GIS contains multiple layers of data; each layer is made up of a number of elements. Each feature in the layer has two aspects to it, its geographical coordinates (but it could be also a polygon, polyline or polypoint) and the data associated with it. GIS store data about layers in database files, with each record in the file referring to a feature in GIS. Actually, NetLogo integration with GIS means shape-file integration while they use the same shape-file; NetLogo is used to read the shape-file data. Agents are created using these data and the simulation process. This means that the context creator provides the population. Agents can be created, re-created and destroyed at every simulation step. The interaction with the environment is provided by the shape-file containing GIS data; multiple GIS layers are the projections within NetLogo. Two projections are assumed, one for the GIS data and the other one for the generated network from this GIS data. The context needs these GIS data for the data fields which provides the information from the environment.

In this approach, the environment consists of polygons representing the network of stores and streets. In fact, this network is divided into cells, namely *store* cells and *street* cells. Each cell is identified by its node. For instance, pedestrian agents can move from a *street* cell to an adjacent *store* cell. Cells containing store information are not always strictly adjacent, for example. In a GIS software package, feature data will be connected to cells of the network and layers will be created. After that, the GIS software package provides the shape-file that will be loaded in NetLogo. This shape-file provides the environmental information. The simulation run can be performed. Each cell in the network has a node. An agent is located in a node on the underlying representation and can move on the implicit generated network to other nodes. Strictly speaking, it does not follow a cellular automata approach because an agent moves from node to node and is situated randomly in the cell related to that node.

The test ground is the inner-city center of Eindhoven. The simulation will be performed on a part of this city center, particularly on a section of the city center. Fig. 2 shows the cellular grid of this segment; Fig. 3 shows a possible population of pedestrian agents in a part of this segment, and Fig. 4 shows the nodes in this part of the segment.

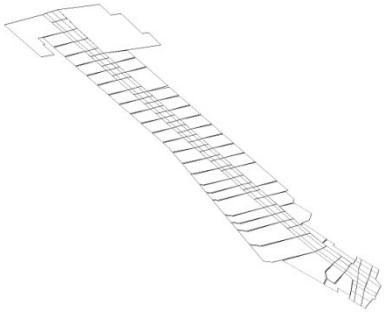


Figure 2. Segment of as section of the inner-city center of Eindhoven.

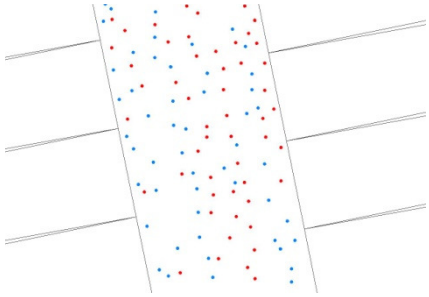


Figure 3. Population with agents in a segement part.

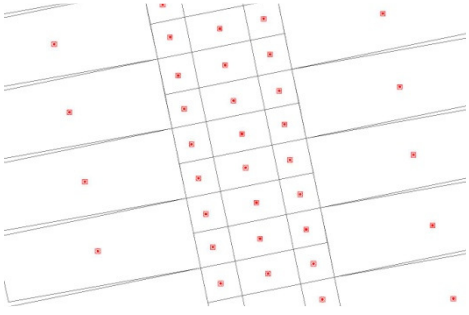


Figure 4. Nodes in this segment part.

 ISBN: 978-1-61208-308-7

TABLE I. POLYGON FEATURES OF THE NETWORK

Identification	Type	Category	Description	Priority
12002	0	5	McDonalds	1.00
12004	0	3	Etos	1.05
11001	0	1	C&A	2.30
13001	1	0	C&A	0.00

Table I shows a number of features as part of the provided MapInfo GIS database; each cell in the network includes these features such as an identification number, type indicating store cell or street cell, and category indicates store category (for example category number 3 represents a health&body store). This database is used by the data fields that are used in NetLogo for environmental information for pedestrian agents perceiving this environment. Priority means the proportion of visits; for example a priority of 2.3 means that 2.3% of the visits are intended to visit that store.

The pedestrian agent model system simulates which shopping activities are conducted by pedestrians, where (destination choice), when (choice of timing), for how long (duration), and which route is used for implementing the activity agenda (route choice). All these activities influence pedestrian's positions in the simulation run. Data collecting efforts are needed to calibrate the agent model system for the test ground: a survey that is a sample of respondents who are asked about their activity agenda. This survey includes questions for pedestrians who have completed their visit to the city center and ask them about the nature of their completed activity patterns (which store, for how long, sequence, and route). Also, a survey was conducted about pedestrian's awareness of stores and signaling intensity of stores as well as the visit of a store and the completion of activities.

IV. PLANNED AND UNPLANNED STORE STOPS

Borgers and Timmermans [7] used Monte Carlo simulation for the incorporation of the numbers of stops and the sequence of planned stops/purposes, because in their opinion the concept of multi-stop, multi-purpose behavior is relevant for understanding pedestrian behavior. According to this line of thought, we assume an activity agenda includes a number of planned and unplanned store visits that can also be considered as a number of non-impulse and impulse store visits.

Every pedestrian agent receives at its introduction in the simulation a pedestrian scenario. This pedestrian scenario includes besides general characteristics like gender, age, companionship also familiarity with the city center, motivation, time budget, and activity agenda. After a store visit the activity agenda will be rescheduled. The number of planned and unplanned store visits is determined by a Monte Carlo simulation.

For this purpose, data from visitors to the city center of Eindhoven are gathered by interviewing them about their motivation and the stores they visited. They are also asked about successful visits and which of them were planned and unplanned; 402 routes are identified.

The findings from the collected data of the number of planned and unplanned visit stops show a skewed distribution. The skewed distributions are different depending of gender, age category and motivation. We often need a skewed distribution where probability densities below and above the mean are distributed differently. In this case, we assume, by analogy with multiple stops, a Gamma distribution.

The probability density function of the Gamma distribution is given by:

$$f(x;k,\theta) = x^{k-1}\frac{e^{-x/\theta}}{\theta^k \Gamma(k)} \quad (1)$$

where, k is the shape parameter (k>0) and θ is the scale parameter (θ>0). Both k and θ will be positive; they are derived from the skewed normal distribution of the number of (planned) stops from their data collection depending on gender, age category and motivation. The shape parameter k is derived from the skewness of the skewed normal distribution (see Fig. 5 for an example) and the scale parameter θ is derived from the mean and k; these parameters are given by:

$$k = \left(\frac{2}{skewness}\right)^2 \ , \ \theta = \frac{mean}{k} \quad (2)$$

Table II shows the values of the parameters of the Gamma probability distribution for different motivation, gender and age category. From the collected data, for the age category was only possible to distinguish between over 55 and less than 55 years.

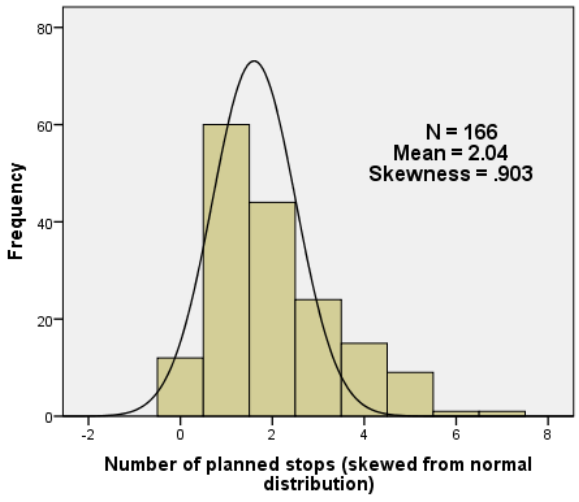


Figure 5. Number of planned stops; Goal Oriented motivation, female & age<55.

 ISBN: 978-1-61208-308-7

TABLE II. PARAMETER VALUES GIVEN DIFFERENT CATEGORIES

Category	Parameter	Motivation				
		Goal oriented		*Leisure oriented*		*No spec. intention*
		Num. stops	Plan. Stops	Num. stops	Plan. Stops	Num. stops
Man						
< 55	k	2.079	2.419	11.973	2.356	4.655
	θ	.957	.662	.279	.845	.421
≥55	k	2.773	1.821	64.515	28.597	1.333
	θ	.721	.890	.035	.050	1.252
Female						
<55	k	4.331	4.906	711.11	9.467	110.80
	θ	.623	.416	.006	.261	.028
≥55	k	1.897	1.174	31.210	95.181	5.642
	θ	1.270	1.678	.138	.025	.507

Striking is the distinction between the number of stops and the planned number of stops. The more goal oriented the less there are unplanned stops.

The Gamma inverse function $G(p)$, which is the inverse cumulative distribution function, is given by (3). Given a random number p from a uniform distribution in the interval (0, 1), the value of $G(p)$ has a Gamma distribution with parameters k and θ. That means, given a number p on the x-axis provides the number of stops on the y-axis; where real values are rounded to integer values.

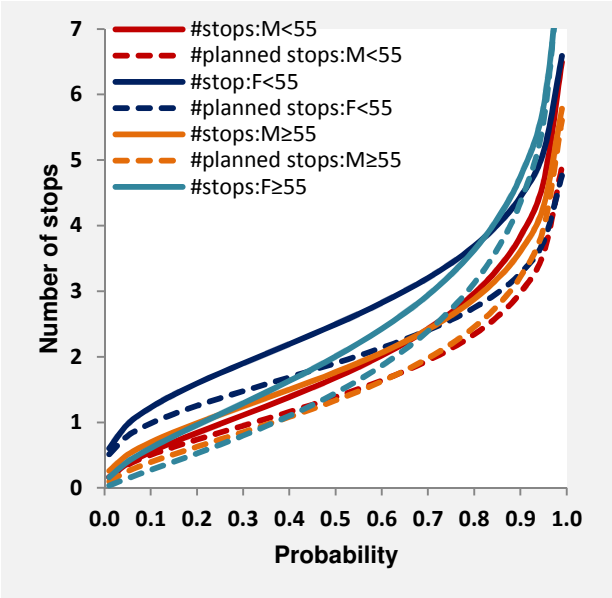


Figure 6. $G(p)$ for Goal Oriented motivation, different gender and age category; Number of (planned) Stops vs. Probability.

$$G(p) = F^{-1}(p; k, \theta) = \{x: F(x; k, \theta) = p\}$$

where (3)

$$p = F(x;, \theta) = \frac{1}{\theta^k \Gamma(k)} \int_0^x t^{k-1} e^{-x/\theta}$$

Fig. 6-8 show the $G(p)$ distribution for respectively *goal oriented* orientation, *leisure oriented* orientation and *no specific intention* orientation with respect to the number of (planned) stops.

Also there is a distinction for gender (male, female) and age (<55, ≥55 years). The number of unplanned stops can be derived from the calculation of the number of stops minus the number of planned stops.

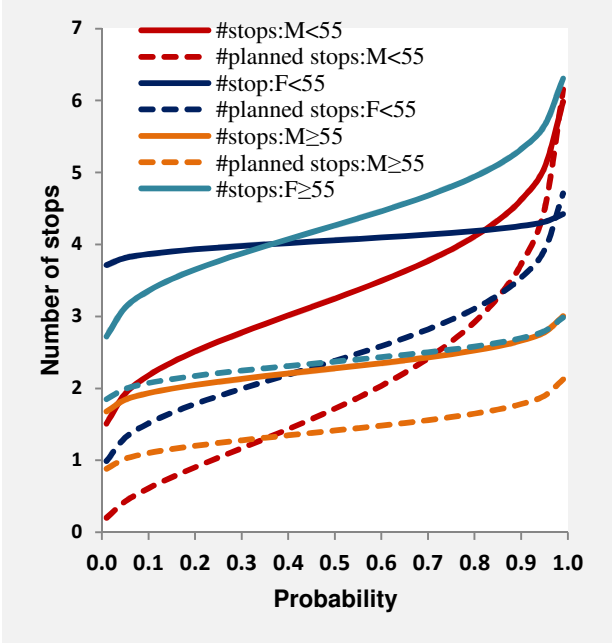


Figure 7. $G(p)$ for Leisure Oriented motivation, different gender and age category; Number of (planned) Stops vs. Probability.

 ISBN: 978-1-61208-308-7

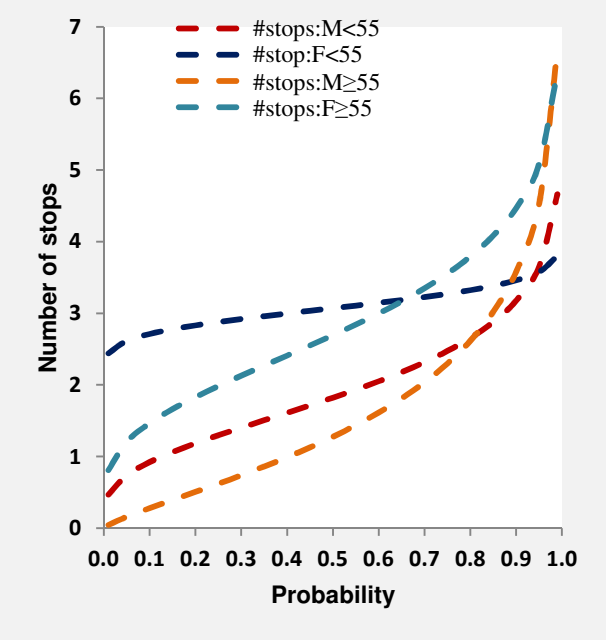


Figure 8. *G(p)* for No Specific Intention motivation, different gender and age category: Number of Stops vs. Probability.

What is clear form Fig. 6-8, that goal oriented motivation more corresponds to a planned route. Leisure oriented motivation more corresponds to an unplanned route, and if one has no specific intention then one has an unplanned route and no planned stops.

V. Discussion and Future Directions

In this paper, we presented only a small part of a simulation platform context for performing the pedestrian movement simulation, namely planned and unplanned visits. Pedestrian behavioral principles were briefly mentioned, and the pedestrian movement simulation setup as well as simulation process was briefly discussed, shown in the activity diagram of Fig. 1. In the current state of the pedestrian agent model system, which is the simulation, all data for the pedestrian movement simulation were collected. The findings about the number of planned and unplanned visits to a store were presented. This gives the number of planned and unplanned stops for the activity agenda in the simulation process. The framework for processing agent-based pedestrian activity simulations will be implemented and all the data will be integrated in the agent model system. This is a step by step process. At this moment, the signaling intensity of a store is represented by a priority of visiting the selected regarding store; and the store visits are split up in planned and unplanned visits.

The pedestrian model system will be tested in a 2D environment, because we want to validate the basic principles. Also, pedestrian agents move from node to node and are situated into the cells related to those nodes. They are situated randomly in those cells, but if the cell is completely occupied by other pedestrian agents, they cannot move to that cell. This approach reduces the complexity of the simulation by ignoring collisions and with that collision detection. These certain characteristics of the system make the simulation feasible because computer power is less binding. If all the parts are implemented, validation of the pedestrian model system will be performed. The data collection will be split up into two parts and the results of the separate simulation experiments will be compared.

Future developments should make the pedestrian agent model suitable for a 3D environment with lifelike virtual persons. In that case, the pedestrian agent movement will be realized from cell point to cell point considering collision detection. Finally, this will result in a virtual environment of a real situation, populated with virtual persons and a real person (user) moving amongst these virtual persons. A user can assess an environment that has high reality content. Preferences of users can be collected and the utility of a proposed situation can be estimated where appropriate. With this approach, it is possible to gain a deeper insight into the activity behavior of city center visitors and thus in the pedestrian flows in city center environments, even for those that do not exist yet.

References

[1] W. Ali and B. Moulin, “How artificial intelligence agents do shopping in a virtual mall: a ‘believable’and ‘usable’ multi-agent based simulation of customers’ shopping behavior in a mall”, in Canadian AI, LNAI 4013, L. Lamontagne and M. Marchand, Eds. Berlin: Springer-Verlag, 2006, pp. 73-85.

[2] J. Dijkstra, A.J. Jessurun, H.J.P. Timmermans, and B. de Vries, “A framework for processing agent-based pedestrian activity simulations in shopping environments”, Cybernetics and Systems, Vol. 42, No. 7, 2011, pp. 526-545.

[3] J. Dijkstra, A.J. Jessurun, and H.J.P. Timmermans, “Simulating pedestrian activity scheduling behavior and movement patterns using a multi-agent cellular automata model”, in Proceedings of the Transportation Research Board Conference, Washington, January, 2002.

[4] M. Bierlaire, G. Antonioni, and M. Weber, “Behavioral dynamics for pedestrians”, in Moving through Nets: the Physical and Social Dimensions of Travel, K.W. Axhausen Ed. Elsevier Science Ltd., 2005, pp. 81-105.

[5] W. Zhu and H.J.P. Timmermans, “Cut-off models for ‘go-home’ decision pedestrians in shopping streets”, Environment and Planning B: Planning and Design, Vol. 35, No. 2, 2008, pp. 248-260.

[6] R. Najlis and M.J. North, “Repast for GIS”, in Proceedings of the Agent 2004 Conference on Social Dynamics: Interaction, Reflexivity and Emergence, C.M. Macal, D. Sallach and M.J. North Eds. Chicago, Illinois, 2004, pp. 225-260.

[7] A. Borgers and H.J.P. Timmermans, “City center entry points, store location patterns and pedestrian route chocie behavior: A micro-level simulation model”, Socio-Economic Planning Sciences, Vol. 20, 1986, pp. 25-30.

 ISBN: 978-1-61208-308-7

Pricing the Cloud: An Adaptive Brokerage for Cloud Computing

Philip Clamp and John Cartlidge
Department of Computer Science
University of Bristol
Bristol, UK
Email: phil@clamped.me.uk, john@john-cartlidge.co.uk

***Abstract*—Using a multi-agent social simulation model to predict the behavior of cloud computing markets, Rogers & Cliff (R&C) demonstrated the existence of a *profitable* cloud brokerage capable of benefitting cloud providers and cloud consumers alike. Functionally similar to financial market brokers, the cloud broker matches provider supply with consumer demand. This is achieved through *options*, a type of derivatives contract that enables consumers to purchase the *option*, but not the *obligation*, of later purchasing the underlying asset—a cloud computing virtual machine instance—for an agreed fixed price. This model benefits all parties: experiencing more predictable demand, cloud providers can better optimize their workflow to minimize costs; cloud users access cheaper rates offered by brokers; and cloud brokers generate profit from charging fees. Here, we replicate and extend the simulation model of R&C using *CReST*—an open-source, discrete event, cloud data center simulation modeling platform developed at the University of Bristol. Sensitivity analysis reveals fragility in R&C's model. We address this by introducing a novel method of Autonomous Adaptive Thresholding (AAT) that enables brokers to adapt to market conditions without requiring *a priori* domain knowledge. Simulation results demonstrate AAT's robustness, outperforming the fixed brokerage model of R&C under a variety of market conditions. We believe this could have practical significance in the real-world market for cloud computing.**

***Keywords*—CReST; simulation; cloud computing; brokerage**

I. INTRODUCTION

Cloud computing is the latest step change in the delivery of computing services as a utility—a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources. Migration to the Cloud involves users moving the location of local compute infrastructure to the network, thereby reducing costs commonly associated with managing hardware and software assets, and gaining from the economies of scale enjoyed by cloud providers [1], [2].

The term "cloud computing" encapsulates both the applications delivered *"as a Service"* and the underlying hardware and software infrastructure in the ultra-large scale data centers that make the concept viable [3]. This infrastructure is commonly known as a *Cloud* and can be *public*, *private*, or a *hybrid* of the two, while a service application delivered to end users is often referred to as *Software as a Service* (SaaS), *Platform as a Service* (PaaS), or *Infrastructure as a Service* (IaaS), depending on which level of the software stack is provided. SaaS typically describes end user applications that are accessed remotely over the internet and includes ubiquitous software applications such as GoogleMail, Facebook, and Twitter. IaaS describes lower-level applications that offer users access to the underlying cloud hardware via a virtualization layer. Typically, for IaaS, users purchase Virtual Machine (VM) instances that are configured with an operating system and offer access to virtual CPU, RAM, and hard disk storage. These VMs can then be configured by the user to provide the specific functionality required. From the user's perspective, VM instances are exactly the same as their own physical hardware accessed remotely. Finally, at the intermediate level between SaaS and IaaS, PaaS offers a suite of software tools and interfaces—a *platform*—upon which users can build and integrate their own software applications. Currently, the clear trend of providers offering ever more bespoke infrastructure products, means that the distinction between PaaS and IaaS is no longer clear (for example, Amazon Web Services' (AWS) RDS Database instance). However, for clarity, in this paper, when we consider cloud resources, we refer to IaaS VM instances and *not* the higher-level software applications (Facebook, Twitter, etc.) that are built on top.

The *on-demand* delivery model for cloud computing resources offers a variety of benefits for business consumers [3]. The ability to start and stop VM instances almost instantly, when required, gives enormous flexibility and scale-out opportunities. In addition, businesses no longer need to invest capital resources in purchasing the often underutilized compute infrastructure needed to cover peak business demand; including all additional costs such as support staff and maintenance [3]. However, the on-demand pricing model is not necessarily ideal for cloud providers, as they attempt to adhere to strict Service Level Agreements in the face of fluctuating demand. If providers could accurately forecast future resource demand, then they would have the opportunity to reduce costs by optimizing electricity purchases, engineering staff, and hardware utilization, etc.

At present, most providers offer a *fixed price* model where VM instances are purchased for a fixed time period (*reserved instances*), or billed per hour of usage (*on demand*). Some providers, e.g., AWS, offer an alternative *spot price* tariff that varies in real-time based on current supply and demand [4]. However, of these methods, only long-term reserved instances (maximum 36 months) aid the provider in capacity planning. Several alternative pricing models have been proposed in academic research, most notably involving *derivatives contracts*, such as (European) *options* [5]. Options contracts involve the payment of an up-front fee that gives the buyer the legal right, but not the obligation, to purchase a resource for an agreed strike-price on some later delivery date [6]. These types of financial instruments are commonly used in financial commodities markets where their underlying assets range from wheat and oil, to a suite of complex financial products.

In their investigation into cloud computing pricing models, R&C used an agent-based simulation model to explore the possibility of a cloud computing services broker delivering

 ISBN: 978-1-61208-308-7

derivative contracts to provide both cheaper resources to consumers *and* aid providers in predicting future usage [4]. They invariably found that not only was it possible to do this, but that in addition the broker was able to generate a significant profit. R&C's result has the potential to significantly impact the delivery and pricing of cloud services. As the market in cloud resources matures and becomes more standardized, the promise of a *federated cloud*—where cloud users can migrate between providers seamlessly—will theoretically allow resources to be traded as a commodity; eradicating existing concerns of vendor lock-in. In turn, this will open opportunities for brokers to enter the market, acting as intermediary *market makers* between users and providers. In such a scenario, R&C's result could have practical as well as academic significance. In this paper, we attempt to replicate and extend the work of R&C. We show that R&C's results are sensitive to model parameter settings and require *a priori* information to maximize profitability. By introducing a novel adaptive learning process, we offer a robust solution to this problem, enabling the broker to automatically maximize profit under a range of market conditions.

This paper is organized as follows. In Section II, we introduce the cloud brokerage model [4] used by R&C to demonstrate the possibility of a profitable broker acting as a third-party mediator between cloud users and cloud providers. In Section III, we briefly introduce CReST—a cloud simulation platform that we use for our empirical simulations—and detail our experimental assumptions and configuration. We then perform three sets of experiments. Firstly, in Section IV we replicate the work of R&C [4] to verify the validity of our simulation model design. Then, to test the robustness of the conclusions drawn by R&C, in our second set of experiments (Section V) we perform a sensitivity analysis on R&C's model. Subsequently, having demonstrated the sensitivity of R&C's optimal threshold value, θ_{opt}, we extend the brokerage model of R&C by introducing a novel method for automatically adapting θ (AAT) during run-time. Our third and final set of experiments (Section VI) demonstrates the performance of AAT under a variety of market conditions. We show that AAT is able to automatically find θ_{opt} under a variety of market conditions with no *a priori* information. Finally, in Section VII we conclude that AAT is a significant, robust extension to R&C's model and one that may have practical significance in the real-world market for cloud computing resources.

II. Background: R&C's Brokerage Model

Typically, the role of a broker is to facilitate the matching of supply and demand in a market. Brokerage services primarily generate profit by charging commission fees, and/or *making the spread* by buying at a lower price and selling at a higher price. In the cloud brokerage model of R&C [4], the broker aims to make a profit by purchasing long-term advanced obligations on resources (36 month reserved instances), and repackaging them as 1 month options contracts that they sell at a higher price to users.

The brokerage model of R&C consists of two stages: (1) each month, the broker takes orders from clients for future resource needs by selling options, and determines how many reserved instances to purchase; (2) in the following month, clients can request instances from the broker by *exercising* their options. If the broker has capacity available from previously purchased reserved instances, they can sell it on to users at a profit. Otherwise, the broker must purchase additional (more expensive) on-demand instances from the provider to fulfill the obligation of the client.

R&C's brokerage model follows a pricing structure that was initially developed at HP Labs by Wu, Zhang, and Huberman (WZH) [5]. The WZH model financially rewards clients that reveal the *true likelihood* that they will utilize a resource in the future. Each month, every client, i, estimates his own probability, p_i, of using a resource in the following month. Clients then submit their estimation, p_i, to the broker in order to purchase a resource option. In the following month, the client is charged $Used(p_i)$ if the option is exercised (i.e., if the resource is used) and $Unused(p_i)$ if the option is not exercised (i.e., if the resource is not used), such that:

$$Used(p_i) = 1 + \frac{k}{2} - kp_i + \frac{kp_i^2}{2} \quad (1)$$

and

$$Unused(p_i) = \frac{kp_i^2}{2} \quad (2)$$

where $k = 1.5$ [5]. If users choose instead to purchase resources directly from the provider, they will expect to pay Op_i, where O is the on-demand cost of a one-month instance (in the original model, $O = 2$ [5]). We can consider this contract as an options model if the broker charges clients $Unused(p_i)$ to purchase the option contract and then a further charge of $Used(p_i) - Unused(p_i)$ in the following month if the option is exercised (if the resource is used). The model can be calibrated to real-world prices by multiplying $Used(p_i)$ and $Unused(p_i)$ by a *cost factor* [4]. It has been proven that this pricing model encourages users to truthfully submit their honest estimate of resource usage, p_i [5].

Each month, once the broker has sold options contracts (and has thus received probability, p_i, estimates from clients), the broker must decide whether or not to purchase additional long-term (36 month) reserved instances from the provider. If the broker has previously purchased enough reserved instances to cover the predicted demand, Σp_i, no further instances are purchased. However, if the broker does not own enough reserved instances to cover expected demand, additional reserved instances are purchased using the following algorithm [4]. Firstly, the broker observes historical resource demand, $\boldsymbol{H} = [h_{t-36}, \ldots, h_t]$, over the previous 36 month period, and compares against the future resource capacity, $\boldsymbol{F} = [f_t, \ldots, f_{t+36}]$, (the number of reserved instances owned) over the forthcoming 36 month period. Using a simple forecasting mechanism that assumes future demand will equal previous demand lagged 36 months, the broker then calculates an *expected deficit profile*, $\boldsymbol{D}$, for each forthcoming month by subtracting historical demand, $\boldsymbol{H}$, from future capacity, $\boldsymbol{F}$, for each month, such that:

$$\boldsymbol{D} = \boldsymbol{F} - \boldsymbol{H}. \quad (3)$$

For each resource required, the *Marginal Resource Utilization (MRU)* is the proportion of months in $\boldsymbol{D} > 0$. The *MRU* estimates the fraction of life ($months/36$) an additional reserved instance is likely to be utilized over the next three years, based on historical demand. Brokers then use a threshold, θ, to determine whether or not to purchase a new 36-month

 ISBN: 978-1-61208-308-7

reserved instance. If $MRU > \theta$, the broker buys a new instance, estimating that it will be used in enough months to make a profit. Alternatively, if $MRU < \theta$, the broker does not purchase a new instance, estimating that it will be underutilized and that purchasing on-demand monthly instances, when necessary, will be more profitable. Each month, the broker delivers 1-month access to reserved instances to clients that exercise their options. If the broker does not have the capacity to fulfill client demand, they purchase additional on-demand instances directly from the provider. In general, the monthly purchase cost of on-demand instances is greater than the monthly cost of 36-month reserved instances. R&C demonstrated that this model can generate broker profits while also benefitting users and providers [4]: users access cheaper monthly resource costs and providers sell a greater proportion of 36-month reservations, aiding in capacity planning to reduce provision costs. For more detailed description of R&C's brokerage model, we refer the reader to [4].

III. Simulation Methodology

The Cloud Research Simulation Toolkit (CReST) was developed at the University of Bristol to address the need for a robust simulation modeling tool for research and teaching of data center management and cloud provision. CReST is a stand-alone application, written in Java, and is freely available open source under a GNU General Public License v3.0 [7]. Although alternative tools exist, CReST has a unique feature set (see [8]) that enables simulation at multiple abstraction levels: from physical hardware, energy usage and thermal flows within a DC, to networked infrastructure and the virtualization layer of application services supporting dynamic user demand. For details on the architecture of CReST, refer to [8].

For all experiments reported in this paper, we use CReST as the cloud simulation platform. CReST is designed as a set of coupled *modules* that can be independently switched on or off depending on the level of abstraction required. Here, to optimize simulation performance, we disabled several of the lower-level physical infrastructure modules, such as the *Thermal* module that tracks air-flow in the data center. The active modules used in all of the brokerage simulations that we perform include: *Brokerage*, *Pricing*, *Events*, *Services*, and *Simulation*. This enabled us to efficiently run experiments that simulate decades of time, without compromising on the abstraction level needed. All CReST code used to run the experiments performed here, and associated Python scripts used for data analysis and visualization, are available to download in version 0.3.0 of CReST [7].

The parameter space used for all experiments, unless otherwise stated, are detailed below:

- **Running Time:** Each simulation lasts 276 simulated months. This time period is determined by the available demand data utilized by R&C (refer to Fig. 1).
- **Number of User Agents:** Following R&C, we set the number of agents that demand resources to 1000.
- **Pricing:** Prices for cloud computing instances in the real world undergo continual change due to underlying factors such as hardware costs and competition. For the R&C replication experiments (Section IV), we follow the same pricing scheme as in [4]. In later experiments (Sections V and VI), we use real-world prices charged by AWS.
- **Reservation and Learning Period Length:** R&C explored 12 and 36 month reservations and demonstrated similar results, but increased broker profits for 36 months [4]. Here, we use only 36 month reservations.
- **Cost Factor:** The WZH charging model [5] is based on reservations with a cost of 1 or 2 and therefore needs to be scaled in order to simulate AWS pricing. In R&C's previous work, the cost factor, C, has varied (i.e., 35 [4] and 60 [9]). In Section IV, we use a cost factor of $C = 35$ to replicate R&C. Then, in Section V, we explore the sensitivity of R&C's model by varying this cost factor.
- **Demand Profiles:** Following [4], to simulate *realistic* demand for virtual machines, we consider four demand profiles generated using real demand data over the period 1988-2011 for a variety of IT-related industries. This data set was collated by Owen Rogers, using the UK Office for National Statistics' database of Non-Seasonally Adjusted Index of Sales. Fig. 1 displays the four demand profiles that we label using R&C's terminology: *Rapid Growth* (top-left), *Steady Growth* (top-right), *Recession & Recovery* (bottom-left) and *Steady* (bottom-right). These data were supplied to us by Owen Rogers to enable us to perform a strict replication of R&C's experiments [4], [9]. For further details on the collection and rationale of data, refer to [4].
- **MRU Thresholds:** In Sections IV and V, we explore a range of thresholds, θ, to determine the optimal (most profitable) value, θ_{opt}, under a variety of market conditions. In Section VI, as an extension to R&C's model, we introduce AAT, a novel technique that automates the selection of θ during runtime.

Each experiment was repeated 30 times to enable statistical hypothesis testing of the results. All code used for experiments detailed in this paper is available to download in CReSTv0.3.0 at https://sourceforge.net/projects/cloudresearch/.

IV. Replication of R&C's Brokerage Model

In [4], R&C use an exhaustive search to determine the optimal MRU thresholds, θ_{opt}, for each of the four markets shown in Fig. 1. They show that θ_{opt} varies between markets and that, when using θ_{opt}, the broker maximizes the profit. They further show that all values of $\theta < 1.0$ generates a profit for the broker in all markets, even when $\theta = 0$; i.e., in the trivial case where the broker will *always* purchase an additional reserved 36-month instance whenever there is a new unit of expected demand. When $\theta = 1.0$, the broker will *never* purchase a reserved instance, hence profits are always 0.

In this section, we replicate the model of R&C as closely as possible in order to: (1) determine whether R&C's results are repeatable; and (2) verify and validate our CReST implementation of R&C's model. To perform this replication, we exhaustively tested a subset of MRU thresholds, $0.0 \leq \theta \leq 1.0$, to determine the profitability of each strategy in each of the

 ISBN: 978-1-61208-308-7

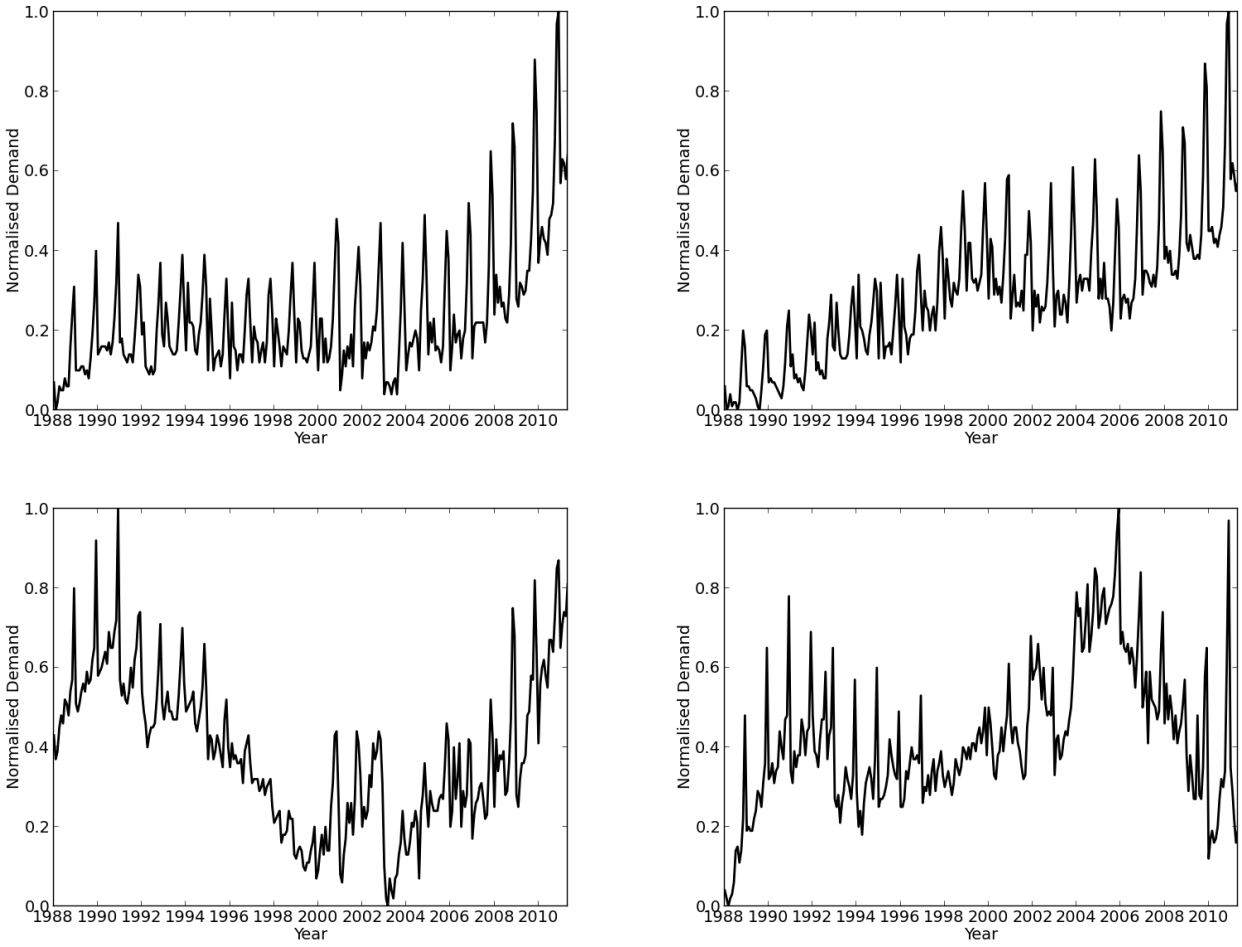


Fig. 1. Normalized demand profiles for the period 1988-2011, labeled: *Rapid Growth* (top-left); *Steady Growth* (top-right); *Recession & Recovery* (bottom-left); and *Steady* (bottom-right). For details, refer to [4].

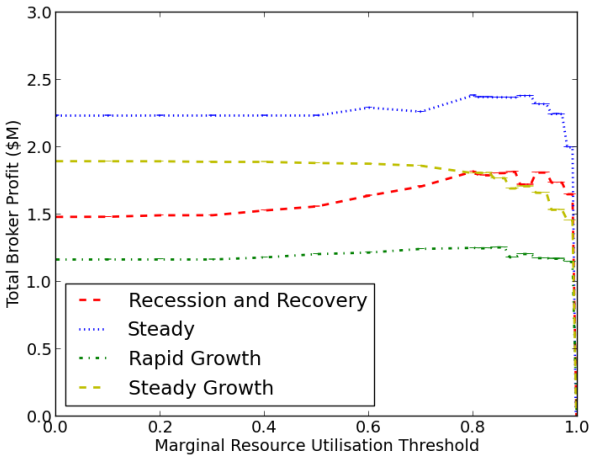


Fig. 2. Total broker profit in $Millions (mean $\pm 95\%$ CI, 30 runs) for each market across different thresholds, θ, using 36 month reserved instances. The resolution of θ between 0.0-0.8 is 0.1 and between 0.8-1.0 is 0.01.

four markets shown in Fig. 1. Results are plotted in Fig. 2. To reduce the search space, eleven θ thresholds were initially tested, such that $\theta \in \{0.0, 0.1, \ldots, 1.0\}$. Performing these simulations using 4 market profiles and repeating each trial 30 times meant a total of $11 \times 4 \times 30 = 1320$ simulation runs. Then, having noticed that the turning point for many of the profit curves in Fig. 2 was in the region of 0.9, an additional set of runs were performed at a resolution of 0.01, such that $\theta \in \{0.81, 0.82, \ldots 0.89, 0.91, 0.92, \ldots 0.99\}$. This led to additional $18 \times 4 \times 30 = 2160$ simulation runs.

In Fig. 2, we see broker profits (mean of 30 runs $\pm 95\%$ confidence interval displayed using vertical bars) for each market, plotted as a function of θ. For *Steady* (blue dots), *Recession & Recovery* (red dash), and *Rapid Growth* (green dot-dash) markets we see broker profits increase with θ until a turning point in the region $\theta \approx 0.9$. However, in the *Steady Growth* market (yellow dash), profits gradually fall as θ rises, until $\theta \approx 0.8$, after which profits rapidly decline. For all markets, when $\theta = 1.0$ brokers make no profit (as expected). Further, for all markets, brokers make a profit for *all* values in the range: $0.0 \leq \theta < 1.0$. These results are *qualitatively* similar to those published by R&C [4].

Table I presents a detailed quantitative comparison of results against the original results of R&C [4]. For each market, we tabulate: (1) the optimum threshold value (θ_{opt}); (2) the

 ISBN: 978-1-61208-308-7

TABLE I. COMPARISON OF BROKER PROFITS ($MILLIONS) ACROSS MARKETS. R&C'S ORIGINAL RESULTS [4] ARE PARENTHESIZED.

Market	θ_{opt}	36 Month Reservations Profit ($M)		
		$\theta = 0$	$\theta = \theta_{opt}$	$(\theta_{opt} - \theta_0)\%$
Rapid Growth	0.84 (0.72)	1.17 (1.15)	1.26 (1.27)	7.7% (10.4)
Steady Growth	0.00 (0.00)	1.89 (1.85)	1.89 (1.85)	N/A (N/A)
Recession & Recovery	0.80 (0.80)	1.48 (1.48)	1.82 (1.80)	23.0% (21.6)
Steady	0.91 (0.82)	2.23 (2.22)	2.38 (2.45)	7.1% (10.4)

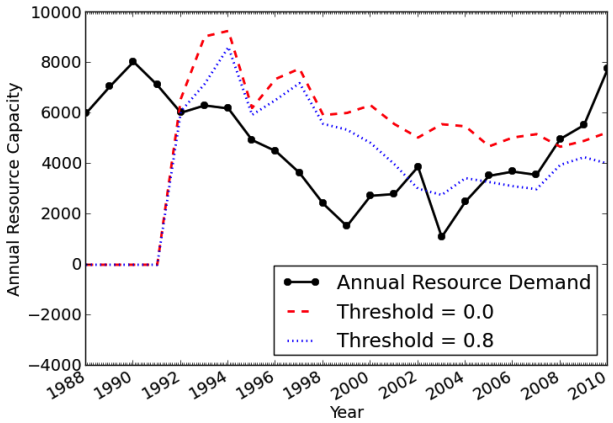


Fig. 3. Annualized broker-owned resources versus demand in a Recession & Recovery market. When $\theta = 0.8$ (blue-dot), the broker's resource purchases more closely track demand (black-line) than when $\theta = 0.0$ (red-dash).

mean profit for brokers that *always* purchase an additional reserved instance ($\theta = 0$); (3) the mean profit for brokers that use the optimum MRU threshold ($\theta = \theta_{opt}$); and (4) the percentage difference in profit between brokers using the optimum threshold value and brokers that always purchase a new instance, i.e., the percentage difference between the previous two columns ($\theta_{opt} - \theta_0$). Values in parentheses are the values obtained by R&C [4]. We see that there is a strong quantitative similarity. All profits are within 5% of the values presented by R&C (indeed, most are within 2.5%). Furthermore, for the optimal threshold values, θ_{opt}, two are identical (*Steady Growth* and *Recession & Recovery*), one is within 10% (*Steady*) and one is within 20% (*Rapid Growth*). As shown in Fig. 2, the *profit gradient* is very shallow in *Rapid Growth* markets (green dot-dash), meaning that profit is relatively insensitive to θ, hence this is the market that we would expect the most discrepancy in results. Overall, we believe that these results demonstrate a strong *quantitative* replication of R&C.

In Fig. 3, we plot the annual broker-owned resource capacity against market demand for two example simulation runs in a *Recession & Recovery* market with MRU thresholds $\theta = 0.0$ (red dash) and $\theta = \theta_{opt} = 0.8$ (blue dots). We see that the optimal θ value (blue dots) more closely tracks actual resource demand (black line), resulting in a greater utilization of purchased 36-month reserved instances. When the broker *always* buys additional instances (red dash), brokers end up purchasing too much capacity, which goes largely underutilized—the area bounded by the red (dash) and black lines from above and below, respectively. This figure demonstrates how tuning the value of θ can enable broker capacity to more closely match user demand, thus maximizing utilization and ultimately maximizing profits. In the majority of markets (*Steady Growth* is the obvious exception), the optimal thresholds, θ_{opt}, tend to be relatively high, falling in the region > 0.8 (and in R&C's original results, in the region > 0.72). This suggests that it is more risky for the broker to purchase a significant number of reserved instances that go underutilized, than it is to purchase fewer and risk buying more expensive on-demand instances. This is not true in the *Steady Growth* market ($\theta_{opt} = 0.0$), where it is *always* beneficial to buy an additional instance since continual market growth guarantees resource utilization.

TABLE II. BROKER PROFITS USING CURRENT AWS PRICING.

Market	θ_{opt}	36 Month Reservations Profit ($M)		
		$\theta = 0$	$\theta = \theta_{opt}$	$(\theta_{opt} - \theta_0)\%$
Rapid Growth	0.4	1.50	1.51	0.67%
Steady Growth	0.1	1.93	1.93	0%
Recession & Recovery	0.6	2.19	2.21	0.91%
Steady	0.0	2.52	2.52	N/A

In this section, we have demonstrated that the cloud brokerage results of R&C are repeatable and verified that our replication of R&C's model using the CReST simulation platform is valid. In the following section, we perform a sensitivity analysis on the model to test the robustness of R&C's results.

V. SENSITIVITY ANALYSIS OF R&C'S BROKERAGE MODEL

In this section, we perform a sensitivity analysis of R&C's brokerage model to determine the robustness of results. In the previous section, we observed that the optimal MRU threshold, θ_{opt}, varies with market demand profile. Here, we analyze the sensitivity of θ_{opt} to other model parameters: (a) resource prices; (b) cost factor; and (c) demand variance.

A. Sensitivity to Provider's Resource Pricing

Here, we update the pricing of resources to reflect the current pricing tariff used by AWS (March 2013):

- Monthly on Demand = $46.80
- Up-Front Reserved = $250.00
- Monthly Reserved = $13.68

We repeated the experiments from Section IV using the pricing tariff presented above. All other configuration parameters were unchanged, including the prices the broker charges clients (the cost factor). Results are presented in Table II. We see that across all markets the optimum threshold, θ_{opt}, is *lower*. Further, the additional profit gained by using the optimal threshold, θ_{opt}, rather than the zero threshold, $\theta = 0$, is much smaller, less than 1% in all markets (final column). This result demonstrates that θ_{opt} is sensitive to the provider's pricing tariffs. In the scenario simulated here, the broker has lower purchase costs (AWS's prices have fallen since R&C's original model). However, the broker does not pass these savings on to users. Hence, the broker's profit in each market increases (compare Table I with Table II). At the same time, the *risk* of purchasing a reserved instance that will be underutilized is lowered. Thus, across all markets θ_{opt} falls.

 ISBN: 978-1-61208-308-7

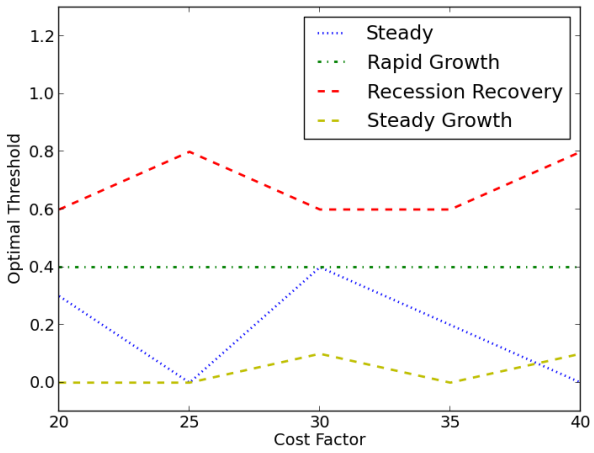


Fig. 4. Optimal thresholds, θ_{opt}, as a function of cost factor.

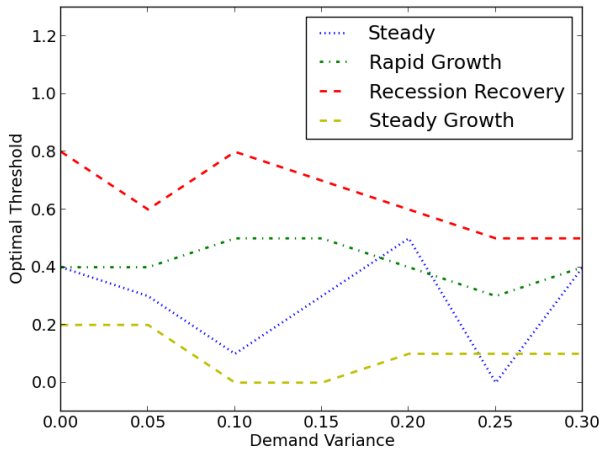


Fig. 5. Optimal thresholds, θ_{opt}, as a function demand variance.

B. Sensitivity to Broker's Pricing

Here, we examine the effect of varying the prices that a broker charges his clients. We control this by varying the *cost factor*, C (refer to Section II). As one would expect, the cost factor variable is directly related to broker profits, with higher cost factor producing higher profits.

Fig. 4 shows the response of θ_{opt} to changes in C. We see that in all markets, apart from *Rapid Growth*, θ_{opt} is sensitive to C and that this relationship is nonlinear.

C. Sensitivity to Variation in Demand

Here, we examine the effect of adding variance (noise) to the market demand profiles presented in Fig. 1. Results are presented in Fig. 5. We see that, in all markets, θ_{opt} is sensitive to variation in the demand profiles and that this relationship is nonlinear.

We have demonstrated that θ_{opt} is highly sensitive to the provider's pricing tariff, to the broker's pricing tariff, and to variation in demand. This confirms that the selection of an appropriate θ_{opt} value for the broker is a nontrivial task. Therefore, we propose that the value of θ should be dynamically adapted in real time in response to contemporaneous market dynamics. In the following section, we propose a novel method for such autonomous adaptive thresholding (AAT) and empirically test its utility. For all experiments, unless otherwise stated, we use the latest AWS pricing tariff presented in Section V-A. We also use a cost factor $C = 30$, selected to preserve the ratio between provider pricing and broker pricing as used in the original brokerage model of R&C.

VI. Extension of R&C's Brokerage Model: AAT

The evident sensitivity of the threshold parameter and its intrinsic contribution to the overall performance of the model presents a complication for the application in real world scenarios. Selecting the optimal θ value enables the broker to balance its asset exposure to the providers in a favorable manner, ultimately reducing risk and maximizing profits. The WZH Model leverages the data of past events in order to hedge risk appropriately. However, due to the nature of its operating environment it is not known *a priori* if the market will continue to follow the same pattern. Up to this point, the experiments conducted have been based on real world past data - however, the inherent unpredictability and vicissitudes of the world markets could render forecasts made on previous demand meaningless. A market shock where demand for a resource in the community suddenly alters, perhaps caused by a new entrant to a market, could lead to the broker operating with a suboptimal threshold parameter, leaving it risk exposed in the number of reservations currently owned. Doubtlessly, therefore it would be advantageous for θ to be automatically updated to reflect the current market circumstances during operation. Here, a versatile technique is presented that enables the broker to autonomously update θ online.

A. Formulation of AAT

The AAT mechanism utilizes the Widrow-Hoff delta rule [10] to streamline the threshold selection between iterations (each month) of the hedging process. The delta rule is a general learning method that has been shown to be effective in a variety of domains, such as Algorithmic Trading [11] and coevolutionary optimization [12]. The delta rule is one of the simplest rules in Machine Learning, forming the basis of both adaptation algorithms [13] and reinforcement in classifier systems [14], [15]. The delta rule attempts to minimize the error between a real system output and a target output determined by some domain-specific proxy. Using the projected reservation utilization as a proxy, AAT updates the θ value in each reservation stage of the model through the minimization of the error between the current threshold and the determined target. If there is no error between the system output and the desired output, then no learning takes place. Conversely, when there is an error, the system values update to reduce this error. The approach can be described with the following set of equations (the notation used is borrowed from [12], which in turn followed from [11]).

Let A_t be the actual output at time t and A_{t+1} be the actual

 ISBN: 978-1-61208-308-7

output on the following time step.

$$A_{t+1} = A_t + \Delta_t \tag{4}$$

where

$$\Delta_t = \alpha(T_t - A_t). \tag{5}$$

Δ_t is the product of a learning rate (α) and the difference between the actual output at t (A_t) and the target output (T_t).

If the target value remains constant, A_t will converge to T_t at the rate determined by α. However, a moving target can cause A_t to oscillate around the target value. In order to dampen the oscillations, an additional variable known as the *momentum* term (μ) can be introduced, transforming (5) to:

$$\Delta_t = \mu\Delta_{t-1} + \alpha(1-\mu)(T_t - A_t). \tag{6}$$

The delta rules expressed above form the basis of the update rule for the MRU threshold. However, as with [12], the target threshold required at each time step is actually unknown and therefore needs to be derived from the data available to the broker. An additional associated variable in the form of a normalized version of the projected resource utilization rate is used, denoted τ. Remembering that a lower θ (close to 0) encourages the purchasing of more reservations, while a higher θ (close to 1) encourages purchasing fewer reservations, τ can be determined:

$$\tau = \frac{reservationsOwned}{summedDemand + 1} \tag{7}$$

where 1 is added to the denominator for cases of no demand.

The rationale for this approach lies with the ultimate aim of the broker to maximize profit through the constant full utilization of the reservations owned, in which case the more expensive on-demand instances would not be purchased and reservations would not go unused. The choice is not without its complications, however. For instance, if the broker owns a relatively large number of reservations, say 100, and the demand for reservations is low, for example 10, the target becomes $\frac{100}{10+1} \approx 9.1$. This is clearly not a suitable target threshold as it exceeds the maximum value of θ considerably. The proposed solution for this involves normalizing the outputted value (see (8)) by keeping track of the largest recorded raw target and normalizing the values between 0 and 1. In this particular example, if 9.1 was the largest seen so far, it would be normalized to 1. If a raw target of 10 had been seen in a previous month, it would be normalized to 0.9, *et cetera*. We normalize τ such that:

$$\tau = \frac{\tau - minTarget}{maxTarget - minTarget} \tag{8}$$

where $minTarget$ and $maxTarget$ are updated over time to determine the relative value of τ. Then, letting θ_t and θ_{t+1} be the threshold at time t and $t+1$, respectively and substituting in τ as the target value, we derive the following AAT formulation from (4) and (6):

$$\theta_{t+1} = \theta_t + \Delta_t \tag{9}$$

TABLE III. HIGHEST RANKING (μ, α) PAIRINGS ACROSS MARKETS

μ	α	Avg. Rank (440 max)
0.7	0.05	400
0.8	0.85	393.75
0.1	0.8	387.25
0.45	0.8	377.5
0.4	0.3	371.25

where

$$\Delta_t = \mu\Delta_{t-1} + \alpha(1-\mu)(\tau - \theta_t) \tag{10}$$

and $\Delta_0 = 0$. The three parameter settings must all fall within the range: $0 \leq \tau,\ \alpha,\ \mu \leq 1$.

B. Selecting Robust AAT Parameters

The reader will notice that AAT introduces new variables to the brokerage model. The value of τ is calculated during run-time using (7) and (8). However, the broker must select parameter values for α and μ. Here, we aim to determine AAT parameter settings that work well *out of the box* under a range of market conditions. This configuration should then enable the broker to maximize profit under a range of market conditions, by self-adapting θ over time in response to variation in demand. In this way, the broker no longer needs to determine θ using *a priori* knowledge of the market they are operating in, thus enabling a more *robust* brokerage model.

To determine appropriate AAT values, we trialed a range of values for $0 \leq \alpha, \mu \leq 1$ (at resolution 0.05), in a variety of market conditions. Table III shows the average ranking of pairwise (μ, α) combinations across the full series of trials. We see that $(\mu, \alpha) = (0.70, 0.05)$ consistently performs well and generates the most profit across all markets. Thus, we use these values to configure AAT for the remainder of experiments performed here, and suggest this configuration as suitable for using the AAT brokerage model *out of the box*. We test the robustness of this configuration in each market, to observe:

1) Convergence behavior: does $\theta^{AAT}_{t\to\infty}$ converge to θ_{opt}?
2) Initialization sensitivity: does the starting threshold value, $\theta^{AAT}_{t=0}$, affect the convergence behavior?
3) Profitability: how does AAT compare with the known static θ_{opt} for each market?

Three starting thresholds were tested: $\theta^{AAT}_{t=0} \in \{0, 1, \theta_{opt}\}$. Each experimental configuration was repeated 30 times.

Fig. 6 shows the yearly mean threshold value, θ, generated by AAT in the *Recession & Recovery* market. It can be clearly seen that, under each condition, the value of θ quickly converges toward $\theta_{opt} = 0.8$, but equilibrates slightly higher. This demonstrates good convergence behavior and insensitivity to the starting value $\theta^{AAT}_{t=0}$. In other markets, AAT convergence is also insensitive to initial conditions (figures not shown, see [16] for more details). However, in other markets, AAT tends to converge to a value $\theta^{AAT}_{t\to\infty} > \theta_{opt}$. Hence, AAT tends to be more *conservative* than the static method, purchasing fewer VM instances than θ_{opt}. Table IV tabulates the profitability of AAT in each market, compared with the profitability of the static threshold, θ_{opt}. We see that, in each market, AAT performs well against the static θ_{opt}, at worst generating 1.64% *less* profit (*Steady* market, $\theta^{AAT}_{t=0} = 1$), and at best generating

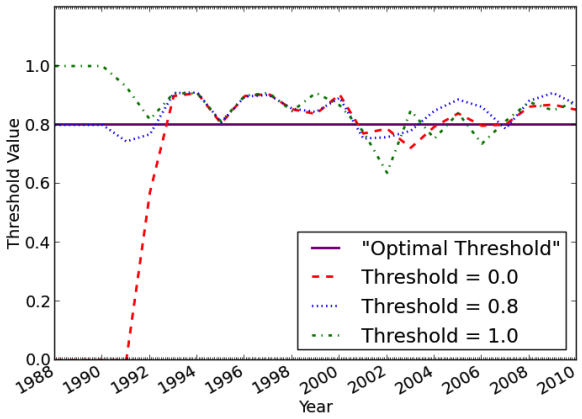


Fig. 6. Yearly mean threshold value, θ, generated by AAT in the *Recession & Recovery* market. Under each starting condition, $\theta_{t=0}^{AAT} \in \{0, 0.8, 1\}$, AAT equilibrates near the optimum threshold value, $\theta_{opt} = 0.8$.

TABLE IV. PROFITABILITY ($M) OF AAT ACROSS MARKETS

Market	Mean Profit ($M) Using Different Configurations			
	Static θ	AAT		
	$\theta = \theta_{opt}$	$\theta_{t=0}^{AAT} = 0$	$\theta_{t=0}^{AAT} = 1$	$\theta_{t=0}^{AAT} = \theta_{opt}$
Rapid Growth	1.088	1.0765 (-1.06%)	1.0789 (-0.84%)	1.0765 (-1.06%)
Steady Growth	1.377	1.367 (-0.73%)	1.362 (-1.09%)	1.367 (-0.73%)
R & R	1.600	1.610 (+0.63%)	1.594 (-0.38%)	1.614 (+0.88%)
Steady	1.764	1.739 (-1.42%)	1.735 (-1.64%)	1.783 (+1.08%)

1.08% *more* profit (*Steady* market, $\theta_{t=0}^{AAT} = \theta_{opt}$). Since this spread of profits is very close to that achieved by the static θ_{opt}, we can conclude that across all markets, AAT: (1) converges toward the known optimal value θ_{opt}, or a more *conservative* value greater than θ_{opt}; (2) is largely insensitive to initial conditions, $\theta_{t=0}^{AAT}$; and (3) can compete with the known static optimum value, θ_{opt}. Since AAT requires no domain knowledge and no *a priori* optimization in each market, we therefore conclude that AAT is a robust extension to the static thresholding technique introduced by R&C. Although we have shown AAT to be largely insensitive to initialization, as a simple heuristic, we suggest initializing AAT to $\theta_{t=0}^{AAT} = 0.5$. This should minimize the average distance to the market optimum, θ_{opt}, and hence should accelerate time to convergence and increase profit.

C. Market Shocks

Here, we perform a final set of experiments to test the utility of AAT when there is a *market shock*, such that market demand suddenly changes from one profile to another. Market shocks occur in real markets when there is a rapid change in demand, perhaps caused by a new market entrant (e.g., see [17] for a discussion on adapting to market shocks). By testing AAT in shocked markets, we aim to simulate more realistic market dynamics. For these experiments, we use the values $\alpha = 0.45$ and $\mu = 0.55$. These were shown to perform well during a series of preliminary experiments.

Fig. 7 shows threshold values, θ^{AAT}, over time (red dash) in one simulation run of a market that is initially a *Recession & Recovery* market and then *shocked* to become a *Rapid Growth* market. The optimal static threshold value, θ_{opt}, is represented

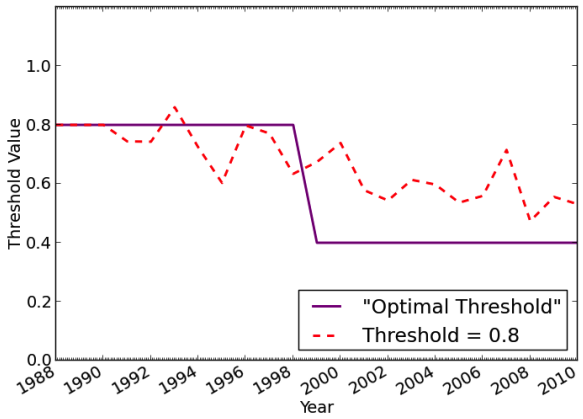


Fig. 7. Market shock from *Recession & Recovery* to *Rapid Growth* market.

by the purple line. We see that $\theta_{opt} = 0.8$ while the demand profile is *Recession & Recovery* and then falls to $\theta_{opt} = 0.4$ while the demand profile is *Rapid Growth*. Initialized with $\theta_{t=0}^{AAT} = 0.8$, we see θ^{AAT} fluctuate around $\theta_{opt} = 0.8$ during the *Recession & Recovery* market phase, and then decline during the *Rapid Growth* market phase, tending to a value of $\theta \approx 0.5$. This value is greater than $\theta_{opt} = 0.4$, but much lower than the optimum value in the *Recession & Recovery* market. This figure illustrates AAT adapting θ appropriately when the market is shocked. However, in other experiments, AAT is not so well behaved (results not shown, refer to [16]). Overall, we conclude that in markets that are shocked, AAT offers advantages over the static method employed by R&C, which is unable to adapt. Yet, results are preliminary and we believe that AAT should be further refined in order to improve the performance. To achieve this, one method that could be employed is "computational steering" [18]; where a computational system is manually *steered* by a human pilot during run time. Unlike a fully autonomous system that is preconfigured and then left to run in isolation with no further human intervention, a computational steering approach to adaptive thresholding would enable the broker to *steer* the AAT parameters over time as market dynamics change. In this way, computational steering enables human input to the system that is otherwise difficult, or *impossible*, to operationally define, such as a domain expert's *tacit knowledge*, or *intuition*.

The market shock experiments reveal the importance of the early stages of reservation hedging for the broker's overall performance. As reservations are a long-term (36-month) investment and since the broker cannot see into the future, there is little that can be done in the short term to circumvent a situation where the broker suddenly owns significantly more or less reserved instances than required. The reader should note that R&C's MRU threshold, θ, controls the proportion of months that the broker is prepared to accept an estimated resource deficit. This is calculated on a monthly basis based on a three year history of demand data. Hence, when a market shock occurs, the MRU technique is negatively disrupted as the previous demand data becomes less relevant to future demand forecasts. As a result, R&C's MRU technique becomes weak when the market is shocked. In contrast, AAT attempts to overcome this problem by enabling the broker to adapt the number of reservations purchased depending on the incoming

 ISBN: 978-1-61208-308-7

demand, even if it is historically atypical. However, the model is still constrained by R&C's demand estimation routine: that future demand can be directly forecast from historical demand. In future, we would like to try an alternative demand estimation model, such as the statistical model presented in [19].

VII. Conclusion

We have replicated and extended the cloud brokerage simulation model of R&C using *CReST*, an open-source, discrete event, cloud data center simulation platform developed at the University of Bristol. To our knowledge, this is the first replication of R&C in the literature and we present our work as validation of their model. However, sensitivity analysis has revealed that R&C's brokerage model is sensitive to configuration parameters, such as: the pricing tariff providers charge for resources, the pricing structure brokers charge their clients, and the effect of noise in the market demand profiles. We present this as evidence that R&C's model requires *modification* before it can be practically used in the real world. To overcome this, we have introduced a novel extension to R&C's model that enables the broker to automatically adapt during run-time to maximize profits, without the broker needing to provide *a priori* knowledge of the market demand or other model parameters. We have demonstrated that this AAT technique is able to converge toward the known optimal value in all markets and that it is robust to initial conditions. We present this as evidence that AAT is a practical, robust extension to R&C's model. We believe this could have practical significance in the real-world market for cloud computing.

Acknowledgment

Thanks to Owen Rogers for detailed discussions of his model implementation and for supplying his demand data to enable replication. Primary financial support for John Cartlidge comes from EPSRC grant number EP/H042644/1.

References

[1] B. Hayes, "Cloud computing," Communications of the ACM - Web Science, vol. 51, no. 7, Jul. 2008, pp. 9–11.

[2] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, U.S. Department of Commerce, Tech. Rep. NIST Special Publication 800-145, Sep. 2011.

[3] M. Armbrust et al., "Above the clouds: A berkeley view of cloud computing," University of California, Berkeley, Tech. Rep. EECS-2009-28, Feb. 2009.

[4] O. Rogers and D. Cliff, "A financial brokerage model for cloud computing," Journal of Cloud Computing: Advances, Systems and Applications, vol. 1, no. 2, Apr. 2012, pp. 1–12. doi:10.1186/2192-113X-1-2.

[5] F. Wu, L. Zhang, and B. A. Huberman, "Truth-telling reservations," Algorithmica, vol. 52, no. 1, 2008, pp. 65–79.

[6] S. H. Clearwater and B. A. Huberman, "Swing options: a mechanism for pricing IT peak demand," in 11th Int. Conf. on Computing in Economics & Finance CEF-2005, Washington, D.C., Jun. 2005, pp. 1–21. [Online] http://www.hpl.hp.com/research/idl/papers/swings [retrieved: Aug, 2013].

[7] CReST, "CReST - the Cloud Research Simulation Toolkit," [Online] https://sourceforge.net/projects/cloudresearch/ [retrieved: Aug, 2013].

[8] J. Cartlidge and D. Cliff, "Comparison of cloud middleware protocols and subscription network topologies using CReST, the cloud research simulation toolkit," in 3rd Int. Conf. Cloud Computing & Services Science (CLOSER-2013), F. Desprez et al., Eds. Aachen, Germany: SciTePress, May 2013, pp. 58–68.

[9] O. Rogers and D. Cliff, "Forecasting demand for cloud computing resources: An agent-based simulation of a two tiered approach," in 4th Int. Conf. Agents & Artificial Intelligence, vol. 2 - Agents (ICAART-2012), J. Filipe and A. L. N. Fred, Eds. Vilamoura, Algarve, Portugal: SciTePress, Feb. 2012, pp. 106–112.

[10] B. Widrow and J. M. E. Hoff, "Adaptive switching circuits," IRE WESCON Convention Rec., vol. 4, Aug. 1960, pp. 96–104.

[11] D. Cliff and J. Bruten, "Minimal-intelligence agents for bargaining behaviours in market-based environments," Hewlett-Packard Labs., Tech. Rep. HPL-97-91, Aug. 1997. [Online] http://www.hpl.hp.com/techreports/97/HPL-97-91.pdf [retrieved: Aug, 2013].

[12] J. Cartlidge and D. Ait-Boudaoud, "Autonomous virulence adaptation improves coevolutionary optimisation," IEEE Transactions on Evolutionary Computation, vol. 15, no. 2, 2011, pp. 215–229.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1: Foundations, D. E. Rumelhart and J. L. McClelland, Eds. MIT Press, 1986, pp. 318–362.

[14] S. W. Wilson, "ZCS: A zeroth level classifier system," Evolutionary Computation, vol. 2, no. 1, 1994, pp. 1–18.

[15] ——, "Classifier fitness based on accuracy," Evolutionary Computation, vol. 3, no. 2, 1995, pp. 149–175.

[16] P. J. Clamp, "Pricing the cloud: An investigation into financial brokerage for cloud computing," Master's thesis, Dep. Comp. Sci., Univ. Bristol, UK, July 2013.

[17] S. Stotter, J. Cartlidge, and D. Cliff, "Exploring assignment-adaptive (ASAD) trading agents in financial market experiments," in 5th Int. Conf. on Agents & Artificial Intelligence, Vol. 1 - Agents (ICAART-2013), J. Filipe and A. L. N. Fred, Eds. Barcelona, Portugal: SciTePress, Feb. 2013, pp. 77–88.

[18] S. Bullock, J. Cartlidge, and M. Thompson, "Prospects for computational steering of evolutionary computation," in Workshop Proc. 8th Int. Conf. Artif. Life (ALife-VIII), E. Bilotta et al., Eds. Sydney, Australia: MIT Press, Dec. 2002, pp. 131–137. [Online] http://eprints.ecs.soton.ac.uk/11459/1/Prospects.pdf [retrieved: Aug, 2013].

[19] J. Cartlidge and S. Phelps, "Estimating demand for dynamic pricing in electronic markets," GSTF International Journal on Computing (JoC), vol. 1, no. 2, 2011, pp. 128–133.

 ISBN: 978-1-61208-308-7

Simulating Tree Plasticity with a Functional-Structural Plant Model: Being Realistic in Behavior

Haoyu Wang[2], Jing Hua[1], Mengzhen Kang[1], Xiujuan Wang[1], Philippe de Reffye[3], Baogang Hu[2]
[1] State Key Laboratory of Management and Control for Complex Systems, mengzhen.kang@ia.ac.cn
[2] National Laboratory for Pattern Recognition
Institute of Automation, Chinese Academy of Sciences, Beijing, China
[3] CIRAD, UMR AMAP, Montpellier, France

***Abstract*—Plant plasticity refers the ability of a plant to change its observable characteristics, in response to the environmental changes in its lifespan. We present a method of simulating structural plasticity in trees reacting to different light intensities, pruning policies, competition, and obstacles. The method is based on a functional-structural plant model (FSPM) that simulates two basic underlying processes of plants: development/organogenesis (the formation of plant structure) and growth (expansion of organs biomass production and partitioning). Bi-directional feedback is constructed between these two processes by linking both bud break and biomass partitioning with the internal source-sink ratio of biomass. A secondary mechanism controlling bud break is its local light intensity, by imposing a light distribution in tree canopy, the computational efficiency for which is assured by implementation on GPU. Based on these mechanisms, the virtual trees produce naturally less branches at lower light intensities. In reaction to pruning, the same tree give different shapes as pruning changes the source-sink balance and triggers new branches. Neighboring trees compete for light and lead to different crowns, and the same mechanism can be used to simulate trees grown near buildings. The results show that by constructing the dynamic model describing the underlying development and growth process of trees in cyberspace, the simulated trees can adapt to their virtual environment without need of modifying their geometrical traits. Such property is interesting for simulating landscape, education and interactive training.**

Keywords-Tree competition; Plasticity; GreenLab; Light environment; Bud break; FSPM; Emergent property.

I. INTRODUCTION

As plant is a ubiquitous component in nature, the realistic presentation of a plant in cyber-space is an ever-existing aim in computer science. Differing from other physical objects such as fluids, plant is a living organism that exhibits phenotypic plasticity, which is the ability of changing its phenotypes (observable characteristics) in response to changes in the environment [1]. Such a feature has brought tremendous challenges to plant modelers, especially for those who desire not just visually plausible virtual plants.

Since 1970's, there has been endeavor toward generating virtual plants according to the underlying algorithms [2] [3] [4]. Combined with techniques of computer graphics, visually realistic 3D plants can be generated with reaction to environment or obstacle [5] [6]. However, the very important aspect of plant growth, biomass production and partitioning, is missing in such pure geometrical or structural plant models. This means, the size of each part of plants, such as stem diameter, leaf length, need to be set delicately in order to obtain proper ratio. The wish of having 'live' virtual plant has brought the concept of Functional-Structural Plant Model (FSPM) [7], which simulates two basic underlying processes of plants: development/organogenesis (the formation of plant structure) and growth (expansion of organs biomass production and partitioning).

In this paper, we aim at simulating tree plasticity with a member of FSPM family, GreenLab. Bi-directional feedback is constructed between these two processes by linking both bud break and biomass partitioning with the internal source-sink ratio of biomass. A secondary mechanism controlling bud break is its local light intensity, by imposing a light distribution in tree canopy. The computational efficiency of computing light inception is assured by implementation on GPU. We show in this paper that the virtual trees from this model can respond automatically to different environmental settings, including different light intensities, pruning policies and competition, without the need of manipulating geometrical parameters.

The paper is organized as follows. Related biological concepts are presented in Section 2. Previous works linked to this paper are reviewed in Section 3. We present an overview of our algorithm and the related models in Section 4, while Section 5 presents several simulation results. Conclusion and discussion of the approach are given in the last section.

II. RELATED BIOLOGICAL CONCEPTS

There are two basic common processes in all plants: *development* and *growth*. Plant development, or organogenesis, deals with the creation of branches and plant organs (leaves, flowers, etc.), while growth refers to the complex process of biomass production by photosynthesis and biomass allocation among individual organs. One of the main hypothesis on biomass allocation is source-sink balance.

For tree development, botanists classified 23 types of *architectural models* [8], according to the organization patterns of botanical units in trees. Each architectural model describes common features of many trees, e.g, the simplest structure is Corner model with a single stem structure. The formation of tree architectures is described as a dynamic process which is “the expression of equilibrium between endogenous growth processes and exogenous constraints exerted by the environment” [9]. In this paper, we present trees of Leeuwenberg model, Rauh model and Roux model.

 ISBN: 978-1-61208-308-7

Leeuwenberg model consists of a sympodial succession of equivalent sympodial units. Pinus tree follows Rauh model with rhythmic growth, orthotropic axis with monopodial branching. Roux model are plants with vertical orthotropic trunks and plagiotropic branches that are always horizontal without righting at the end [9].

For *biomass production*, *light interception* can be computed as plant level or organ (leaf) level. Beer-Lambert Law assumes that light attenuates in a canopy as if it is a semi-transparent object [10]. It holds for closed tree stand, and the thickness of the canopy can be quantified by leaf area index (LAI, ratio of total leaf area to its projection ground area). With 3D description of tree structure, another approach is to apply photosynthesis model at leaf level, but it needs simulation of light distribution inside tree canopy. In this work, we test both kinds of photosynthetic models.

For *biomass partitioning* inside a plant structure, the mechanism is less known than the overall biomass production [11]. Hypotheses on partitioning include functional equilibria [12], source-sink regulation [13] and allometric relationships [14]. A special feature differing trees from crops is their ring growths. One well-accepted theory is Pipe Model [15], indicating that the diameter at a certain position of a stem is proportional to the number of leaves above it. In this work, ring growth is based on pipe model, and organ size is the result of source-sink regulation.

Mechanism on *bud activity* regulation is multiple: light, auxin, source-sink ratio of assimilates, etc. [16]. Buds are origin of branches, and their activities affect strongly the tree shape. They can stay dormant, die or give birth to branches. Buds are of different *physiological ages*, representing the vigor of branch that it can bear [9]. For example, in Gingko tree, some buds bear twigs, while others give birth to long branches. Axillary buds are generally physiologically older than their parent branches. *Reiteration* is an exception, referring to such a phenomenon that a branch inherits the same features of its mother branch [8][9]. This is often considered as 'self-similarity' property in trees and has led to application of fractal method in tree construction. Breakout of bud is highly dependent on plant age and environment. Bud behavior can explain many of tree plasticity in reacting to their environment, i.e., changes of size, amount, orientation or color to fit better during their life time. In this paper, we test two mechanisms controlling bud break: external light condition and internal source-sink regulation.

III. RELATED WORK

A. Simulation of Virtual Plants

Generating 3D tree structures started not long after the birth of personal computers [3]. Recursive generative algorithms have been applied by viewing a tree as an explicitly-defined recursive structure [17]. Started by field investigation and mathematical modeling, AMAP methodology integrates knowledge like physiological age, architectural model and bud activities [4][18]. Botanical tree library has been set up for hundreds of trees. Although tree animation can be achieved by above methods, the size of compartments is directly defined by rules but not from photosynthesis and biomass partitioning.

Another approach is to use images or point clouds to reconstruct the 3D-tree by registering their input images or using loosely arranged images [19]. To infer the hidden internal branches, iteration rules such as a particle flow system can be imposed to link the main stem with external twigs [20], which link the rule-based and data-based approach. Diameter of trees can be inferred from data but again there is no 'growth'.

B. Simulating Environmental Effect

Because of its importance, light is often taken as the principle environmental condition. Greene [21] simulated climbing plants on obstacles using voxel automata and light rays. Derived from a standard L-system, Měch et al. [22] developed open L-systems that have been used for simulating tree competition, based on communication between the plant and its environment. Soler et al. [23] simulated light environment in trees with an efficient method. Van Haevre [24] proposed a ray density estimation of the environmental illumination to guide phototropism morphology. In above works, light casts effects on branch removal or bending, but they have no contribution to photosynthetic production of plants, which plays a principle role in tree growth.

C. Functional-Structural Plant Model (FSPM)

LIGNUM [25] is a FSPM dedicated to trees and shrubs that couples L-systems for tree development with an eco-physiological model. L-peach [26] is another FSPM for young peach trees, and it allows the simulation of pruning with empirical description on number of new shoots. AMAPHydro [27] is a branch of AMAP [4], which introduced a hydraulic model for computing biomass production. The computational efficiency of early FSPMs was low, and implementation was usually prone to bugs [7]. GreenLab [28] [29] inherited many concepts from both AMAPHydro and AMAPSim, but the computational efficiency was greatly improved by applying sub-structure method [30], so that it is affordable for complex trees. It is one of few FSPMs that has been calibrated on both crops and trees, by fitting it to multiple biomass data, such as for pine trees [31] and beech trees [29].

D. Control on Bud Break

According to different hypotheses on bud activity control, various methods have been proposed and tested. The first type is based on hormone regulation, which is interpreted as a signaling mechanism [22]. It is often implemented in pure developmental plant model. The second is based on source-sink regulation, with source being provided by photosynthesis and sinks being the demand of organs for resources. By setting that the number of bud break is dependent on the dynamic source-sink ratio, rhythmic branching pattern could be generated [32]; Eschenbach [33] simulated tree structures with gradient according to environmental conditions. Such mechanism can be built only

 ISBN: 978-1-61208-308-7

with functional-structural plant model as there is issue of biomass production (source) and allocation (sink).

The third method is to link directly bud formation to its local light condition. In [34], it was linked to the far red spectrum while [35] supposed that bud fate is decided by light availability. In [22], the branch apices are associated with the radii of interest for collision detection, and a bud will stop growing when there is not enough space available. Such method is commonly used in simulating plant communities where plants compete for resources [35].

In this work, the bud break is controlled by two factors: local light condition and source-sink ratio. Therefore, light can affect bud fate both directly by deciding bud breakout, and indirectly by contributing to the overall source-sink ratio.

E. Controlling Tree Form by Interaction

A challenge of rule-based model is that one may lose the control over the final tree shape. A pure interactive example is Speedtree, but dynamic growth sequence of trees is not easy to construct. Chiba and Ohkawa [36] simulated interactive pruning of tree for designing Pensai trees. Pirk et al. [6] simulated the removal of branches when a tree meets an obstacle or other trees. In the context of tree pruning, intermediate storage of tree structure can be necessary, using format such as multi-scale tree graph [37].

Although it will be nice to introduce procedural brush as in [5] and [38], we focus on the pruning exercise as done on real trees in final shape control. The tree shape is controlled by interactive pruning on trees, which will break the internal source-sink balance and stimulate the breakout of buds, including the dormant buds.

IV. System Overview

As described by Měch and Prusinkiewicz [22], the interaction of a plant with the surrounding environment can be conceptualized as two concurrent processes that communicate with each other, forming a feedback loop of information flow. In our case, besides the external one, there is an internal information flow between plant structure and function; therefore, both light (that affects both the structure by controlling bud formation and function by photosynthetic production) and pruning (that affects the structure) have effect on the plant structure. External modification of environment will naturally trigger the response of plant. The double feedback loops are shown in Fig. 1.

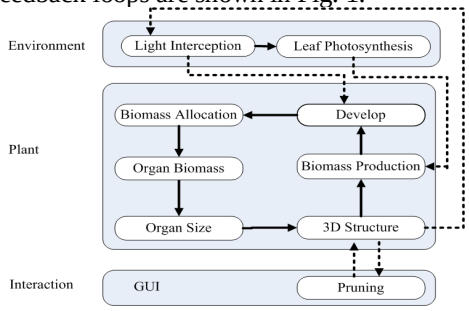


Figure 1. External interaction between plant and environment and internal interaction between plant development and growth.

V. A Functional-Structural Model for Trees

A. Development model

The description of organogenesis in GreenLab is based on the definition of the potential bud production. A simple illustration can be seen in Fig. 2, where each circle presents a bud of a certain Physiological Age (PA) [9]. PA of the main stem (blue) is 1. Each rectangle represents a metamer, a minimal botanical unit that is composed by a node, an internode and its axillary leaves and fruits. Branching structure can be formed by the parallel bud development into growth units (from successive appearance of metamers).

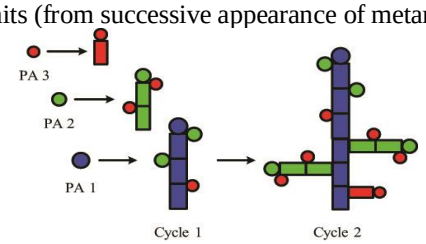


Figure 2. Diagram of plant development model in GreenLab, a deterministic case. Each circle presents a bud.

The parameters that describe the development are expressed as a matrix. For the branching structures of Fig. 2, the number of metamers in a growth unit and PA of the axillary buds are written as follows:

$$M_I = \begin{bmatrix} 0 & 2 & 1 \\ 0 & 0 & 2 \\ 0 & 0 & 1 \end{bmatrix}, N_B = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad (1)$$

where $M_I[1,2]=2$ means that in a growth unit of PA 1, there are potentially two metamers that bear buds of PA 2. $N_B[1,2]=1$ means for such a metamer, the amount of potential buds it carries is 1. Recall that PA of axillary buds is generally higher than its parent stem, except for the case of reiteration. The corresponding structure at iteration cycle 5 is shown in Fig. 3a.

A main feature of the above model is that the botanical axis are organized using the concept of PA. Branches of the same PA share the same parametric settings for development, growth and geometry. Numeric values instead of rewriting rules are needed in designing the topological structure.

B. Reiteration

In this functional-structural plant model, reiteration is simulated by setting the PA of an axillary bud equal to the age of its mother axis. An example in Fig. 3 illustrates this mechanism. Instead of bearing two buds of PA 2, the growth unit of PA 1 contains one metamer with a reiteration bud ($M_I[1,1]=1$), one metamer with a axillary bud of PA 2 ($M_I[1,2]=1$), and one metamer with a bud of PA 3 ($M_I[1,3]=1$). All metamers carry at most one bud ($N_B[1,i]=1, 1 \le i \le 3$).

$$M_I = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, N_B = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad (2)$$

 ISBN: 978-1-61208-308-7

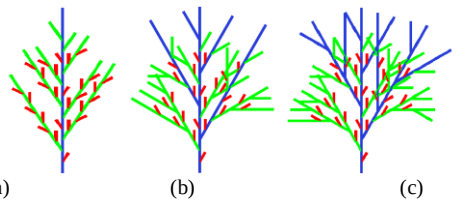


Figure 3. Topological structures at cycle 5. (a) no reiteration, parameters in (1); (b),(c) reiteration of PA 1 and PA 2, parameters in (2), maximal reiteration order being 1 (b) and 2 (c) respectively.

A parameter called maximal reiteration order R_m limits the level of reiteration, which can be understood as the maximal branching order of reiteration. If there is no control, fractal structure will be born. Fig. 3 shows the resulting structure at cycle 5, with R_M=1 (Fig.3b) and R_M=2 (Fig.3c) for stem of PA 1 (blue) and PA 2 (green). No reiteration in PA 3 (red).

C. Bud Break

Bud control is based on two kinds of hypotheses: source-sink balance [32] and light condition [35]. We keep the light distribution model as optional in case that fast simulation is needed. The diagram on bud control can be seen in Fig. 4.

To reach the above goal, when the local light condition allows, the number of axillary buds of PA *q* in a growth unit of PA *p*, $N^{p,q}$, is a function of simulated source-sink ratio; see (3). $Q(n)$ is computed using plant or leaf level photosynthesis model; see (8) and (9). $D_V(n)$ is potential plant demand if all candidate buds break. α^q is a coefficient indicating the dependency of the bud break on source-sink ratio.

$$N^{p,q} = \min\left\{\alpha^q \frac{Q(n)}{D_V(n)}, M_I[p,q]\cdot N_B[p,q]\right\} \quad (3)$$

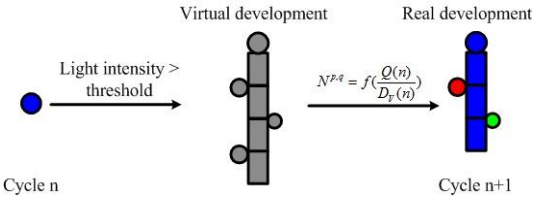


Figure 4. Approach of controlling bud break. Actual bud production is dependent on plant source-sink ratio. The bud production is possible when its local light intensity is above a threshold.

D. Biomass Partitioning

This model is based on the hypothesis of source-sink regulation. All produced assimilates are distributed among the growing organs according to their sink strengths. For trees, it is further hypothesized that the sink for ring growth $D_L(n)$ is dependent on source-sink ratio [29]; therefore, the thickening rate of stems is dependent on tree age and external conditions. The total demand of plant $D(n)$ at cycle *n* is the sum of all sink strengths, see (4).

$$D(n) = \sum_O \sum_p P_O \cdot N_O^p + D_L(n) \quad (4)$$

where N_O^p is the number of organ *O* at PA *p*, whose sink strength is P_O. The parameters of sink strength can be estimated by inverse method from data [29] [31].

In order to decide the fate of bud, at each cycle, a virtual demand $D_V(n)$ is firstly computed by summing up all sink strength from potential buds. A sink strength of a bud is defined as the total sink strength of its potential organs.

$$D_V(n) = \sum_p P_O \cdot N_{Bud}^p + D_L(n) \quad (5)$$

The actual plant demand is computed when the bud behavior is fixed according to $Q(n)/D_V(n)$. The produced biomass is then shared for creation of new metamers and thickening of old stems. For the latter, distribution of biomass is in relation to the number of functioning leaves above each metamer [29].

E. Interactive Pruning

At each cycle, through a GUI, users are allowed to select and remove some leaves, flowers or internodes interactively from the virtual plant. This information is fed back to the simulator before moving forward to next cycle (Fig.1). As there is underlying data structure indicating the topological link between all parts, if an internode is removed, all branches above it are removed. Updating of structure changes both source and potential sink, and consequently the source-sink ratio for next cycle. According to the mechanism of bud control, this can trigger different behaviors of buds. As bud extension takes several cycles to be visibly evident, the effect of pruning is not immediate.

F. Light Interception

In this work, the light intensity around a leaf plays double roles: determining the fate of adjacent bud and the total plant production. To simulate light environment, we emit light rays evenly from a sky hemisphere covering the plant canopy, as in [21]. Each sample ray collides, reflects and decays in the tree canopy. Photon map [39], which is originally used for rendering of a scene, records the information of collision between rays and objects.

The light intensity around a leaf, denoted as E_L, can be estimated from photons in its neighborhood. As in trees, leaves are generally densely distributed inside the canopy, more photons found in the neighborhood of a leaf means more light interception by surrounding leaves, and hence less intercepted light by the leaf. This is an inverse situation to the original photon mapping algorithm [39]. The absolute light intensity E_B is computed as in (6):

$$E_B = (1 - \frac{E_L}{E_L^{\max}}) \cdot (\tau_{\max} - \tau_{\min}) + \tau_{\min} \quad (6)$$

where $E_L^{\max}$ is the maximal value of E_L of all blades. τ_{max} and τ_{min} denote the maximal and minimal light intensity above canopy an inside canopy, respectively, both of which can be measurable by canopy analyzers.

G. Photosynthesis Model

This model concerns on the relationship between Photosynthetic Active Radiation (PAR) intercepted by leaves

and biomass production. We test two kinds of photosynthesis models computed at leaf and plant level, with or without using geometrical information of tree.

1) Leaf level: In this case, the photosynthesis model is applied at organ (leaf) level, according to light intensity computed for each leaf. A generalized light-response curve is used to compute instantaneous assimilation rate of an individual leaf (I, μmol $CO_2 \cdot m^{-2} \cdot s^{-1}$, using a non-rectangular hyperbola [42]:

$$I = \frac{b - \sqrt{b^2 - 4\theta\alpha E_B\beta(I_m + R_d)}}{2\theta} - R_d \quad (7)$$

where $b = \alpha E_B\beta + I_m + R_d$. The physical meaning and empirical values of variables α, β, I_m and R_d are from [40].

Total biomass production of the whole plant is summed from those of individual leaves, as in (8).

$$Q_L(n) = \delta_t\gamma \sum_{i=1}^{N_B(n)} I_i(n)s_i(n) \quad (8)$$

where γ is a conversion coefficient from assimilate to dry mass, δ_t is the duration of a growth cycle (s), $N_B(n)$ denotes the total number of leaves in the plant. $I_i(n)$ and $s_i(n)$ are assimilation rate and the area of i^{th} individual leaf, respectively, the latter being computed iteratively by the model. This metho has the advantage of taking into account the geometrical shape of trees and obstacles, but it is more time-consuming.

2) Plant level: The plant-level photosynthesis model is based on Beer-Lambert law [10]. The thickness of the canopy is quantified by LAI, an important value in evaluating light interception. To estimate LAI for individual trees, in GreenLab, each tree has a characteristic projection area (S_P), which can increase with plant age in the beginning and finally stabilize when the tree canopy closes, see (9).

$$\begin{cases} Q_B(n) = E(n)S_P(1 - \exp(-k\frac{S(n)}{S_P})) \\ Q_B(0) = Q_{seed} \end{cases} \quad (9)$$

where $E(n)$ is a variable representing the plant local environment at growth cycle n; k is a light extinction coefficient to quantify attenuation process of light penetrating into the canopy; $S(n)$ is the total green leaf surface area at growth cycle n; Q_{seed} is the initial biomass. Under certain parameter values, the results from both methods (Q_L and Q_B) fit each other. This method has high computational efficiency as it is not dependent on tree geometrical structure.

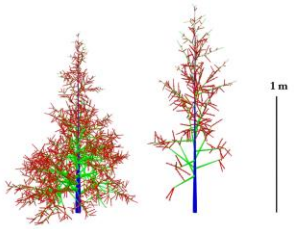
H. GPU+CPU implementation

In case of implementing leaf-level photosynthesis model, there are two performance bottlenecks: ray tracing and light intensity estimation. In ray tracing, bounding volume hierarchies (BVH) and k-d tree are widely used to accelerate intersection computation, collision detection and k nearest neighbor search (KNN). In recent years, these algorithms are implemented successfully on GPU [41][42]. To improve system performance, this part is implemented on GPU. The GPU is Nvidia GeForce GTX560Ti with a 1023MB of dedicated memory.

Simulation of tree development and growth, as well as the interactive pruning, are all implemented in QingYuan software programmed in C++. This kernel of model is still implemented in CPU. Rendering is made in Pov-Ray. All examples in this paper were generated on a desktop computer equipped with Intel(R) Core(TM)2 Quad CPU@2.66GHz with 8GB of memory.

VI. RESULTS

A. Tree structure dependent on source-sink ratio



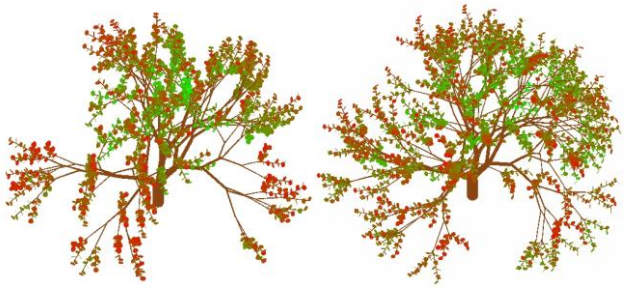
(a) (b)

Figure 5. Effect of α ((3)) on bud break. (a) less dependency on source-sink ratio; (b) more dependency on source-sink ratio.

As the visual output is based on a biophysical model, simply by playing on certain parameters, such as the α ((3)) that controls bud break, one can obtain trees of different complexities, as in Fig. 5. Fewer branches appear at lower α value, and the plant automatically becomes taller. This emergent result is in line with a common practise that people remove side shoots to obtain a tall trunk. This parameter can be used to obtain trees of different cultivars. Different colors in Fig. 5 represent different PAs.

B. Simulating tree plasticity with different light intensities

Using leaf photosynthesis model, tree plasticity under different light intensities above canopy are simulated (Fig. 6). Colors from red to green are used to visually distinguish light intensities inside canopy. Higher light intensity in the right gives positive feedback to bud break out and hence more dense crown, which is visually obvious in Fig. 6b. Notice that excessive light does not mean endlessly branching as this is controlled by potential buds defined in development model. Moreover, the photosynthetic response curve will saturate with the increase of light intensity.



(a) (b)

Figure 6. Trees simulated with low (a) and high (b) light intensities. Colors represent relative light intensity inside canopy, with red and green for high and low light intensity respectively.

 ISBN: 978-1-61208-308-7

C. Simulating tree response to pruning

Realistic tree behavior in reaction to pruning is shown in Fig. 7. The age of displayed trees is 15 cycles, and they were pruned at cycle 9. From left to right, policy of pruning is no pruning, removal of terminal bud of main stem, removal of terminal buds of all stems, and cutting top of main stem, respectively. As less buds are presented after pruning, the sequence, more pruning leads to more branches near the bottom. Trees are taller or shorter, dependent on the time and position of pruning. There is no forced rule indicating number of new shoots after pruning.

Figure 7. Response of trees to pruning. From left to right, policies of pruning are no pruning, removal of terminal bud of PA 1, removal of terminal buds of all branches, and cutting top of main stem, respectively.

D. Simulating tree competition

As plant-level photosynthesis model has high computational efficiency, we first test tree plasticity with the single control of bud break with source-sink regulation. From a predefined seed (a given set of source and sink parameters), the same tree exhibits plastic behaviors in response to different S_P values as in (9); see Fig. 8. Higher density gives smaller, thinner and slightly shorter trees.

Figure 8. Trees simulated under different population densities, from left to right: $S_P = 64$ m^2, $S_P = 16$ m^2, $S_P = 4$ m^2, $S_P = 1$ m^2.

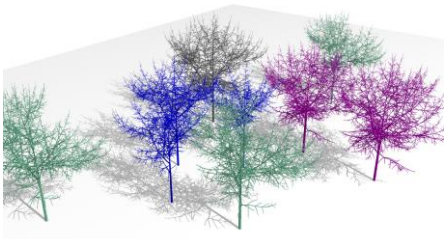


Figure 9. Trees of different crown size in response to their local area (S_P). The trees with same color have the same S_P value.

This way, given the location of each individual tree in a stand, the local occupied area of each tree can be computed and this will automatically limit the size of tree crown from invading other trees. Fig. 9 shows a tree stand where individual trees are distributed randomly. Here no branch collision detection is applied.

Without consideration of local light control on bud break, the simulated tree crown is isotropic. Taking into the secondary mechanism of bud control with light, we simulated two neighboring trees grown up in parallel (Fig. 10). The competition for light started at a certain age of plants. The asymmetrical tree shapes are caused by low light intensity at neighboring area where less buds break out. There are self-pruning in the trees in that old branches fall down automatically.

E. Simulating tree response to obstacle

Figure 11. Effect of pagoda on tree shape.

With simulation of light distribution in canopy, we simulated a tree grown near a pagoda whose buds sense the local light level (Fig. 11). While the standard-alone tree (left) shows full canopy, the tree in the right loses part of branches. This is also visible for the same tree at younger stage (middle).

Fig. 12 shows a nice example on how bud behavior is controlled jointly by light intensity and source-sink ratio. There is no more control on bud break when the tree grows over the roof, its crown recovered by reiteration structure (the top branches are the same).

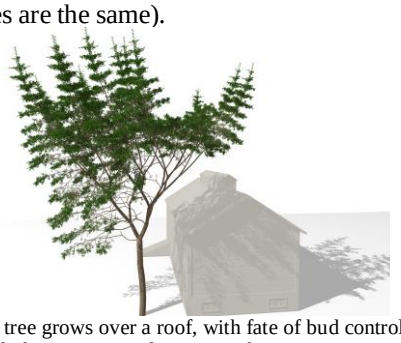


Figure 12. A tree grows over a roof, with fate of bud controlled by light intensity and source-sink ratio.

F. Performance

To evaluate the performance, we compare the time used for computing the tree using plant-level photosynthesis model, leaf-level photosynthesis model on CPU, and leaf-level photosynthesis model on CPU + GPU (Table I). The other modules of biomass allocation and organogenesis remain the same. The computation time for light intensity at each cycle (light model) is very costly for CPU implementation, but the CPU + GPU method improved greatly the efficiency of light calculation, being more obvious for complex trees (177 times for age 25). Total simulation time for tree development and growth reduced greatly accordingly. The computational efficiency is much

 ISBN: 978-1-61208-308-7

Figure 10. Response of trees to pruning. From left to right, policies of pruning are no pruning, removal of terminal bud of PA 1, removal of terminal buds of all branches, and cutting top of main stem, respectively.

higher when plant-level model is applied (<1s), independent on plant age or tree complexity. This means response to planting density or pruning of trees can be achieved in real-time.

TABLE I. PERFORMANCE FOR COMPUTING TREES AT DIFFERENT AGES. TIME FOR LIGHT INTENSITY AT EACH CYCLE (LIGHT MODEL) IS REDUCED BY CPU+GPU (C+G) IMPLEMENTATION.

Tree age	Organs	Plant-level	Organ-level			
		Simulation Times (s)	*Simulation Times (s)*		*Light Model (s)*	
			CPU	*C+G*	*CPU*	*C+G*
10 years	3,860	0.11	31.08	10.6	8.42	1.06
20 years	55,466	0.28	631.37	32.3	257.62	2.52
25 years	547,832	0.49	4,181.00	99.71	931.74	5.26

VII. CONCLUSION

We presented a dynamic biophysical tree model, GreenLab, which simulates tree structure and its plasticity in response to environment (obstacle, density, light) and management (pruning). The adaption of trees is automatic by deciding bud fates from internal and external conditions, without applying deformation or collision detection. The same tree (defined by a set of parameters) can display vastly different structures, because of the power of underlying mathematical model. Our method is affordable to create dynamic tree library for various circumstances. Depending on needs, different combinations of photosynthesis models and treatments can be chosen, with corresponding tree shape and cost. By using GPU programming, computational efficiency is high even when leaf-level light interception and photosynthesis model are conducted.

A limit of our work is that to design a tree based on GreenLab, users should possess some knowledge of botany and eco-physiology. Once the parameters are given for a tree, user can modify environment and interact with the virtual tree, as if it were a living organism, without caring for the internal mechanism of tree growth. It is interesting for scene designer, since putting a virtual tree is similar to putting a seed, and then one can see the interaction and dynamic evolution of the trees. The potential of our work also lies in the possibility of simulating effects of other environmental conditions, such as temperature and CO_2. Users can even have optimal solution of treatment if applying optimal control on the virtual plant.

Our work is similar to some previous works as it also simulates the light environment and plant response. The major difference is that we simulate not only tree development, but also tree growth with an internal feedback mechanism between both processes. Light stimulates not only bud break but also photosynthetic production. Furthermore, we can simulate more complex botanical plant architecture and modify the meristem activities (depending of Q/D). Growth Units are very flexible, which makes the interaction between growth and environment very efficient and faithful.

By means of these simulation methods and models, it could be easily used on education of botany, landscape design, games and so on. What a potential and more valuable aspect is the application on agroforestry. By combining light model and pruning management into one organic whole, we could evidently ameliorate light environment in canopy under different pruning strategy, which reduces pests and thus play a role in improving fruit quality. These virtual experiments provide quantitative standards for fruit cultivation and management. This is the future direction of our research.

Based on current hypothesis chosen in the model, tree plasticity from the potential development pattern, regarding to bud breakout, organ sizes and final tree shape become emergent properties of the model. However, the model is still open to other mechanisms, such as hormone regulation by signal propagation.

REFERENCES

[1] T. D. Price, A. Qvarnström, and D. E. Irwin, "The role of phenotypic plasticity in driving genetic evolution," Proceedings of the Royal Society of London. Series B: Biological Sciences, vol. 270, no. 1523, 2003, pp. 1433–1440.

[2] A. Lindenmayer, "Mathematical models for cellular interactions in development. part i and ii," Journal of Theoretical Biology, vol. 18, 1968, pp. 280–99,300–15.

[3] H. Honda, "Description of the form of trees by the parameters of the tree-like body: Effects of the branching angle and the branch length

 ISBN: 978-1-61208-308-7

on the shape of the tree-like body," Journal of Theoretical Biology, vol. 31, 1971, pp. 331–338.

[4] P. de Reffye, C. Edelin, J. Francon, M. Jaeger, and C. Puech, "Plant models faithful to botanical structure and development," Computer Graphics, vol. 22, no. 4, 1988, pp. 151–158.

[5] W. Palubicki, K. Horel, S. Longay, A. Runions, B. Lane, R. Me˘ch, and P. Prusinkiewicz, "Self-organizing tree models for image synthesis," in ACM Transactions on Graphics (TOG), vol. 28, no. 3. ACM, 2009, p. 58.

[6] S. Pirk, O. Stava, J. Kratt, M. Said, B. Neubert, R. Měch, B. Benes, and O. Deussen, "Plastic trees: interactive self-adapting botanical tree models," ACM Transactions on Graphics (TOG), vol. 31, no. 4,2012, p. 50.

[7] R. Sievänen, E. Nikinmaa, P. Nygren, H. Ozier-Lafontaine, J. Perttunen, and H. Hakulai, "Components of functional-structural tree models," Annals of Forest Science, vol. 57, 2000, pp. 399–412.

[8] F Hallé, R. Oldeman, and P. B. Tomlinson, "Tropical trees and forests: an architectural analysis," New York: Springer-Verlag, 1978.

[9] D. Barthélémy and Y. Caraglio, "Plant architecture: a dynamic, multilevel and comprehensive approach to plant form, structure and ontogeny," Annals of Botany, vol. 99, no. 3, 2007, pp. 375–407.

[10] M. Monsi and T. Saeki, "Uber den lichtfaktor in den pflanzengellschaften und seine bedeutung fur die stoffproduktion," Japanese Journal of Botany, vol. 14, 1953, pp. 22--52.

[11] I. Grechi, P. Vivin, G. Hilbert, S. Milin, T. Robert, and J.-P. Gaudillère, "Effect of light and nitrogen supply on internal c:n balance and control of root-to-shoot biomass allocation in grapevine," Mathematics and Computers in Simulation, vol. 59, no. 2, 2007, pp. 139--149.

[12] H.-L. White, "The interaction of factors in the growth of lemna- xii. the interaction of nitrogen and light intensity in relation to root length," Annals of Botany, vol. 1, 1937, pp. 649--654.

[13] I.-F. Wardlaw, "The control of carbon partitioning in plants," New Phytologist, vol. 116, 1990, pp. 341--381.

[14] G.-B. West, J.-H. Brown, and B.-J. Enquist, "A general model for the origin of allometric scaling laws in biology," Science, vol. 276, 1997, pp. 122--126.

[15] K. Shinozaki, K. Yoda, K. Hozumi, and T. A. Kira, "A quantitative analysis of plant form -the pipe model theory. parts i and ii," Japanese Journal of Ecology 1968, vol. 14, pp. 97--105, 1964, 133--139.

[16] S. Shimizu-Sato and H. Mori, "Control of outgrowth and dormancy in axillary buds," Plant Physiology, vol. 127, no. 4, 2001, pp. 1405--1413.

[17] P. Prusinkiewicz, L. Muvndermann, R. Karwowski, and B. Lane, "The use of positional information in the modeling of plants," in ACM SIGRAPH 2001, Los Angeles, CA, USA, 2001, pp. 289--300.

[18] P. de Reffye, "Modélisation de l'architecture des arbres par des processus stochastiques. simulation spatiale des modèles tropicaux sous l'effet de la pesanteur. application au coffea robusta," Ph.D. dissertation, Université de Paris-Sud centre d'Orsay, 1979.

[19] A. Reche, I. Martin, and G. Drettakis, "Volumetric reconstruction and interactive rendering of trees from photographs," ACM Transactions on Graphics (SIGGRAPH Conference Proceedings), vol. 23, no. 3, July 2004.

[20] X. Chen, B. Neubert, Y.-Q. Xu, O. Deussen, and S. B. Kang, "Sketch-based tree modeling using markov random field," in ACM SIGGRAPH Asia 2008 papers, ser. SIGGRAPH Asia '08.

[21] N. Greene, "Voxel space automaton: modeling with stochastic growth processes in voxel space," Computer Graphics, vol. 23, no. 3, 1989, pp. 175--184.

[22] R. Měch and P. Prusinkiewicz, "Visual models of plants interacting with their environment," Computer Graphics, vol. 30, no. 3,1996, pp. 397–410.

[23] C. Soler, F.X. Sillion, F. Blaise, and P. de Reffye, "An efficient instanciation algorithm for simulating radiant energy transfer in plant models," ACM Transactions on Graphics, vol. 22, no. 2, 2003, pp. 204--233.

[24] W. Van Haevre, F. Fiore, P. Bekaert, and F. Van Reeth, "A ray density estimation approach to take into account environment illumination in plant growth simulation," in Proceedings of the 20th spring conference on Computer graphics. ACM, 2004, pp. 121--131.

[25] A. Lacointe, "Carbon allocation among tree organs: A review of basic processes and representation in functional-structural tree models," Annals of Forest Science, vol. 57, no. 5/6, 2000, pp. 521--533.

[26] M. Allen, P. Prusinkiewicz, and T. M. DeJong, "Using L-systems for modeling source-sink interactions, architecture and physiology of growing trees: the L-PEACH model," New Phytologist, vol. 166, no. 3, 2005, pp. 869--880.

[27] F. de Reffye, P.and Blaise, S. Chemouny, S. Jaffuel, T. Fourcaud, and F. Houllier, "Calibration of hydraulic growth model on the architecture of cotton plants," Agronomie, vol. 19, 1999, pp. 265--280.

[28] H. Yan, M. Kang, P. de Reffye, and M. Dingkuhn, "A dynamic, architectural plant model simulating resource-dependent growth," Annals of Botany, vol. 93, no. 5, 2004, pp. 591--602.

[29] V. Letort, P. H. Cournède, A. Mathieu, P. de Reffye, and T. Constant, "Parametric identification of a functional-structural tree growth model and application to beech trees (fagus sylvatica)," Functional Plant Biology, vol. 35, no. 10, 2008, pp. 951--963.

[30] P.-H. Cournède, M.-Z. Kang, A. Mathieu, H.-P. Yan, B.-G. Hu, and P. de Reffye, "Structural factorization of plants to compute their functional and architectural growth," Simulation. Transactions of the Society for Modelling and Simulation International, vol. 82, no. 7, 2006, pp. 427--438.

[31] F. Wang, M.-Z. Kang, Q. Lu, V. Letort, H. Han, Y. Guo, P. de Reffye, and B.-G. Li, "A stochastic model of tree architecture and biomass partitioning: application to mongolian scots pines," Annals of Botany, vol. 107, no. 5, 2011, pp. 781--792.

[32] A. Mathieu, P.-H. Cournède, V. Letort, D. Barthe`le`my, and P. de Reffye, "A dynamic model of plant growth with interactions between devel- opment and functional mechanisms to study plant structural plasticity related to trophic competition," Annals of Botany, vol. 103, 2009, pp. 1173–1186.

[33] C. Eschenbach, "Emergent properties modelled with the functional structural tree growth model almis: Computer experiments on resource gain and use," Ecological Modelling, vol. 186, no. 4, 2005, pp. 470--488.

[34] J. Evers, J. Vos, X. Yin, P. Romero, P. van der Putten, and P. Struik, "Simulation of wheat growth and development based on organ-level photosynthesis and assimilate allocation," Journal of Experimental Botany, vol. 61, no. 8, 2010, pp. 2203--2216.

[35] J. Hua and M. Kang, "Functional tree models reacting to the environment," in ACM SIGGRAPH 2011 Posters. ACM, 2011, p. 60.

[36] N. Chiba and S. Ohkawa, "Visual simulation of botanical trees based on virtual heliotropism and dormancy break," The Journal of Visualization and Computer Animation, vol. 5, 1994, pp. 3--15.

[37] C. Godin, E. Costes, and H. Sinoquet, "A method for describing plant architecture which integrates topology and geometry," Annals of Botany, vol. 84, 1999, pp. 343--357.

[38] B. Lintermann and O. Deussen, "Interactive structural and geometrical modeling of plants," IEEE Computer Graphics & Applications, vol. 19, no. 1, 1999, pp. 2--11.

[39] H. Jensen, Realistic image synthesis using photon mapping. AK Peters, Ltd., 2001.

[40] T. Givnish, "Adaptation to sun and shadea whole-plant perspective" Functional Plant Biology, vol. 15, no. 2, 1988, pp. 63–92.

[41] C. Lauterbach, M. Garland, S. Sengupta, D. Luebke, and D. Manocha, "Fast bvh construction on gpus," in Computer Graphics Forum, vol. 28, no. 2. Wiley Online Library, 2009, pp. 375–384.

[42] K. Zhou, Q. Hou, R. Wang, and B. Guo, "Real-time kd-tree construction on graphics hardware," in ACM Transactions on Graphics (TOG), vol. 27, no. 5. ACM, 2008, p. 126.

 ISBN: 978-1-61208-308-7

A Non-Modular Modeling and Simulation Approach Based on DEVS for the Forest Fire Spread

M. Hamri
LSIS UMR CNRS 7296
Aix-Marseille university
Marseille, France
Email: amine.hamri@lsis.org

Y. Dahmani
EECE lab
Ibn Khaldoun university
Tiaret, Algeria
Email: dahmani_y@yahoo.fr

***Abstract*—Recently, the modeling and simulation of forest fire spread using discrete event formalisms have been intensively investigated. In this paper, we propose a non-modular approach vs. partial-modular and modular approaches based on Discrete EVent system Specification (DEVS) formalism to reduce exchanged messages between cells and to improve essentially performances of forest fire spread model. Note that the existing DEVS models simulate the forest fire spread for only small scale forests.**

***Index Terms*—forest fire spread M&S; DEVS.**

I. INTRODUCTION

The Modeling and Simulation (M&S) formalisms are used to understand, represent, specify, and analyze the dynamic of systems. Often, these systems are very complex due to the fact that in a small temporal window there is a high number of variables which change values. Consequently, this complexity increases according to the simulation time.

Modeling the system dynamics is a hard task, particularly natural systems in which abstraction of the behaviors should be done to reduce the system complexity. Different methods and techniques were developed in order to improve the formulation of such systems. In the literature, two main categories are distinguished: analytic methods, which are difficult to grasp and M&S ones. Formally, a large variety of behaviors can be formulated mathematically. To learn about systems, we must take into account all involved variables and entities. Although, their dynamics reveals a complex formulation involved by these variables, the corresponding equations are unable to provide accurate results due to the increasing complexity of data.

The M&S is based on an experimental frame. The likeness between experimentation and M&S was the essence of this twinning [1], [2] offering the possibility of predicting the behavior of complex systems. Various approaches were defined to handle the two steps of M&S, depending on time-driven or event-driven.

In the application domain, due to the important damages caused by fire, governments employed the necessary humans and resources to limit these damages and intervention costs of firemen. The scientists on their side have provided efforts to understand and counter at best the forest fires. Consequently, many models were developed to firemen and decision makers to train them and define the efficient strategy.

The mathematical model of Rothermal is one of the viable models for forest fire. It employs a set of continuous variables interrelated (vegetation fuel, wind speed and direction, humidity, etc.), which influence the direction and speed of the forest fire spread. However, the Rothermal model is specific to north America terrains and can not be reused elsewhere.

On the other hand, simulation models are very accessible and easy to use. We find the two categories time-driven and event-driven to M&S of forest fire through the decomposition of forest in the two-dimensional space. The cellular automata were widely used in this field. The spread of the forest fire is based on simple rules executed at each time step. Some of these rules are extracted from the Rothermal model.

In event-driven simulation, Discrete EVent system Specification (DEVS) contributed to this field. The DEVS-Fire model proposed by the authors in [3] combines DEVS formalisms and Rothermal model and the shown results are very interesting. However, despite the use of a heap-based simulation engine to load and simulate only active cells at each simulation cycle and the enhance of neighbor cell ignition process (pre-schedule model vs. on-time schedule one), the corresponding models need an important heap memory and CPU time to execute the behavior of forest fire spread. Consequently, only a small set of active cells are allowed to be simulated, even if the authors note that the model is able to simulate wildfire with large scale.

In order to remedy the lack of such an approach in which each forest cell is modeled with an DEVS atomic model and for each active cell a simulator is invoked to produce the equivalent behavior, we propose to model the whole forest with a unique DEVS atomic model, which describes the fire spread like reality. The paper is organized as follows: Section 2 gives a recall on DEVS M&S and Section 3 discusses our proposal. Section 4 shows the object-oriented design of the new model of forest fire spread. Finally, we conclude on this work and we outline the perspectives.

II. DEVS PRINCIPLES

DEVS is one of the popular discrete event formalisms proposed in 70's by Zeigler [4]. The DEVS M&S framework separates clearly modeling concerns from simulation ones. In fact, DEVS abstract simulator is useful to produce the behaviors of any model that respects the DEVS definitions. On the other hand, DEVS models are reused and coupled

 ISBN: 978-1-61208-308-7

among them to make new DEVS models. Many research and practicable works were realized around this formalism thanks to its powerful expressiveness. This formalism has many extensions: GDEVS [5], Cell-DEVS [6], etc. and applications in different fields: forest fire spread, workflows, etc.

A. DEVS Atomic Formalism

According to the literature on DEVS [7], the specification of a discrete event model is a structure, M, given by:

$M = (X, S, Y, \delta_{int}, \delta_{ext}, \lambda, D)$, where X is the set of the external input events, S the set of the sequential states, Y the set of the output events, δ_{int} is the internal transition function which defines the state changes caused by internal events, δ_{ext} is the external transition function which specifies the state changes due to external events, λ is the output function, and the function $D : S \rightarrow R^+ \cup \infty$ represents the maximum length or the lifetime of a state. Thus, for a given state s, $D(s)$ represents the time during which the model will remain in state s if no external event occurs.

B. DEVS Coupled Formalism

DEVS promotes modular modeling to reduce the complexity of the system to describe. The DEVS coupled strucutre allows to formalize the modeled system in a set of inter-connected and reused components.
$MC = (X_{MC}, Y_{MC}, D_{MC}, M_{d|d \in D}, EIC, EOC, IC, Select)$, where

- X_{MC}: set of external events.
- Y_{MC}: set of output events.
- D_{MC}: set of components names.
- M_d: DEVS model named d.
- EIC: External Input Coupling relations.
- EOC: External Output Coupling relations.
- IC: Internal Coupling relations.
- $Select$: defines a priority between simultaneous events intended for different components.

This formalism is proved by the closure under coupling property, which shows that a DEVS coupled model is a DEVS atomic one. However, the formalism is less useful to describe large-scale systems like cellular DEVS, etc. Although, it seems possible at conceptual level, the corresponding computerized model can not be coded or at least simulated on computer due to the fact that the simulation needs a large heap memory. In [8], the authors show that the closure under coupling can be used to change the DEVS coupled model by its equivalent DEVS atomic. Consequently, the heap memory is reduced to the minimum to load the DEVS atomic model and its simulator, which consists of a root-coordinator and a basic simulator. However, the authors note that the transformation from DEVS coupled to atomic is not conducted automatically and intelligent modeling is recommended. This point constitutes our main goal.

C. DEVS Simulator

The DEVS abstract simulator (see Fig. 1) consists of a root-coordinator, which manages the simulation time, sub-coordinators which dispatch messages according to the specific couplings of the coupled model that attempt to simulate and basic simulators related to atomic models. Each process behaves according to the received messages from parent and child processes.

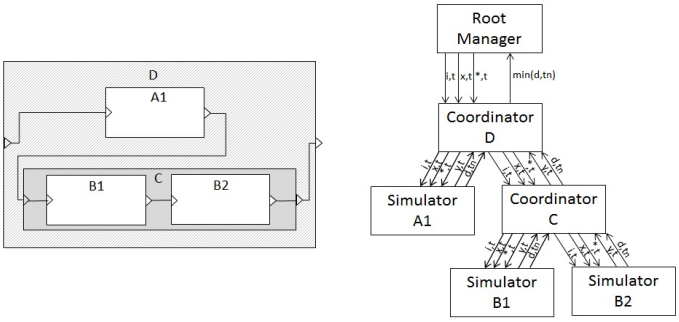


Fig. 1. DEVS abstract hierarchical simulator.

The classic structure of DEVS simulator is a hierarchical one, represented as a tree in which at top level is the root followed by the sub-coordinators created from DEVS coupled structure; then, at low level there are basic simulators related directly to the corresponding DEVS atomic models in order to execute the different functions $\delta_{int}, \delta_{ext}, \lambda$ and D. Fig. 1 illustrates this structure and messages transiting from a process to another.

III. Related works

Many scientific works discuss the forest fire spread and propose models to anticipate the fire direction and calculate its Rate of Spread (ROS). in [9], Weber noted that three categories of such models are developed in the literature: statistical and Markovian models, empirical models like Rothermal and Albini models and Physical models, which consist on reproducing the forest fire spread characteristics with law rules. In the third category, the simulation constitutes an efficient means for M&S of the forest fire behavior. Two paradigms are very popular: Cellular Automata (CA) and Discrete Event Simulation (DES). CA are dynamical models in which the space of cells is a set of states and the time is an integer. The cells are arranged in a two-dimensional space and their shape is often square. Applying CA consists on spreading fire from a burning cell to its neighbors using Moore or von Newman neighborhood. Some works [10] used hexagonal cells to have more realistic simulations. In the field of DES, [3], [11], [12] used DEVS and its extensions Dynamic-Structure DEVS (DS-DEVS), Cellular DEVS, etc. to model and simulate forest fire spread. The main parameter in such models is the ROS, which allows computing delays (times) to ignite neighbor cells according to several cell parameters (fuel, slope, etc.) and weather.

As an example, DEVS-Fire [3] allows to model and simulate the spread and suppression of fire. It consists of three models: Rothermal model to compute the ROS according to natural parameters (wind, terrain, etc.), firefighting model to manage and execute the planned strategy to suppress fire and the

 ISBN: 978-1-61208-308-7

forest fire spread model, which ignites unburned cells and deletes burned ones or saved ones by firemen. The last model is a DEVS coupled one, which describes the state of the forest. It decomposes the forest into equal limited cells (areas) according to two-dimensional space. Each cell is a DEVS atomic model, which has eight neighbors (except those situated on the boundaries) and all together form the forest. Each cell influences its neighbors with events sent out and received via ports, so an external modular coupling is involved. In order to enhance DEVS-Fire performances, DS-DEVS is used to create and delete cells dynamically. This adjustment is motivated by the fact that only a few number of cells are active during the simulation of fire spread. Consequently, for each simulation cycle useless messages like igniting passive cells (burned and noninflammable) cells are ignored and a reduced heap memory is allocated only for active active cells; passive cells will not be loaded. This main advantage is loosen when an important number of cells are active for a simulation cycle (for more details see [13], [14]).

The implementation of the DEVS cellular space of DEVS-Fire following the previous description is known under the modular approach in [15] and OnTime_Schedule model in [3]. The alternative implementation to the modular approach is the partial-modular one [15] (pre_Schedule model [3]). In this implementation, the number of simulation cycles to ignite neighbors of a burning cells is not eight cycles but only one due to the fact that the burning cell sends out the delays for which neighbors should ignite and not the event ignite. Consequently, less messages are exchanged between cells and simulation execution time is reduced.

However, these implementations are heavy by designing forest cells with identical DEVS atomic models to get on memory the cell state and to ignite neighbors on time or by pre_scheduling. In the next section, we propose a new model for the forest fire spread based on DEVS.

IV. New DEVS Model for the Forest Fire Spread

Designing each forest cell in DEVS-Fire with an atomic model has the advantage that the approach remains modular. Consequently, cells could be coupled with other DEVS models. However, the use of external couplings via DEVS coupled leads to a large structure of simulation for cells which have the same behaviors. Although, the model is based on a dynamic structure in which only active cells are simulated; the model is limited to simulate a small set of active cells at the same time in order not to overload the heap memory and not to spend time on creating and deleting active and passive cells respectively in case of a large set of active cells.

A. Our Proposal

In our modeling of the forest fire spread, we model the whole forest decomposed into cells with a unique DEVS atomic model. Each cell is a state variable of this model, holds a DEVS behavior and communicates with its neighbors through an internal couplings. In fact, the communication between cells is done directly inside the model, except events which influence the outside of model. We can model the natural parameters that influence the forest fire spread with other DEVS models coupled with the forest cell model or sensors which keep values from the real world or estimated by the user to update the simulation parameters.

The literature distinguishes three classes of parameters which set the ROS: vegetation type (caloric content, density, etc.); fuel properties (vegetation) and environmental parameters (wind speed, humidity and slope, etc.). The flaming fire evolves mainly according to the direction of the wind, its velocity and the relative humidity. The present model uses two relevant baseline parameters: wind velocity and relative humidity. The humidity influences the wild-land fire behavior by increasing the risk factor. Low relative humidity is an indicator of high fire danger. A dry and powerful wind, associated with a dry ground, enormously increase the fire spread.

Firstly, we identify two main categories of cells:

1) Nonflammable cell: can be a road, a surface of water or just an empty surface. It is designed with a static model.
2) Inflammable cell: is each area from the forest that is sensible to fire. According to its natural parameters and when the event ignite occurs in the corresponding area, the cell transients from the initial state unburned to burning to represent consuming fire; after some duration estimated from natural parameters, the cell goes to state ember to show the fact that small glowing piece of coal or wood in dying fire. Then, the cell reaches the state burned, which is the final combustion process. At this stage, the nonvolatile products and residue were formed when matter is burned (see Fig. 2).

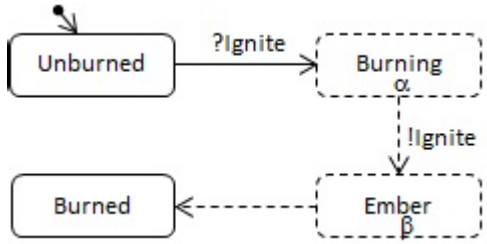


Fig. 2. DEVS behavior for inflammable cell.

Note that the difficulty of this model is the estimation of α and β the lifetime of burning and ember states respectively. In our case, they are functions of wind velocity, humidity, and fuel area.

Other categories could be described, such as areas with risks (houses, plants, etc.) and introduced them into our model. Therefore, we should first describe the behavior of such areas with DEVS and then, we plug them inside the forest cell model.

B. Formal Model of the Forest Fire Spread

The model of the forest fire spread that we propose is a DEVS atomic model, which collapses all cells inside one

 ISBN: 978-1-61208-308-7

model to model forests with large scale and to enhance simulation performances. It is described as follows:

$ForestFireSpread = (X, Y, S, \delta_{ext}, \delta_{int}, \lambda, D)$
$X = \{(ignite, list), (wind, velocity), (humidity, value)\}$
$list$: the list of cells to ignite initially.
$Y = \emptyset$. There is no output event to send out.
$S = Cell \times Wind \times Humidity \times Fuel$

$Cell$ is the set of cell areas represented in the two-dimensional space. Each cell is identified by its position (line, column) and typed according to two categories nonflammable and inflammable mentioned above. Each cell keeps its current state and according to external and internal events state changes will occur to update the state of the concerning cell. Note that each cell has eight neighbors except those situated on the boundaries. $Wind$ is a global parameter. We assume that the wind is uniform for all cells. It can be supposed as a constant or change over time. $Wind$ is described with two parameters $direction(degree)$ and $speed(km/h)$. We choose this description to avoid the linguistic description for wind direction (from north, south, etc.) and to get an exact value. $Humidity$ is also a global parameter that we suppose uniform for all cells. $Fuel$ is specific to each cell. It is used to compute the times of burning and ember of the corresponding cell.

The functions $\delta_{ext}()$, $\delta_{int}()$ and $D()$ are shown in Fig. 3.

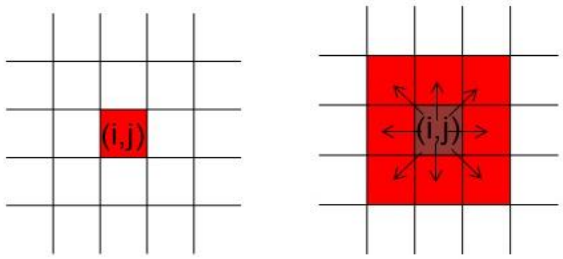
$\delta_{\mathbf{ext}}(\mathbf{s}, \mathbf{e}, \mathbf{x})$
c, c´, c_a : Cell
if (x = ignite){
for each c ∈ $ignite_{list}$
$s_c = \delta_{ext}$ (s_c, e, ignite)
}
if(x = Wind)
update Wind
if (x = Humidity)
update Humidity
recompute the lifetime for each active cell c

$\delta_{\mathbf{int}}(\mathbf{s})$
for each $c \in Cell$\{
if (lifetime(c) = lifetime(Cell))\{
if (burning(s_c))
for each c´ ∈ neighbor(c)
$s_{c'} = \delta_{ext}(s_{c'}$, lifetime(Cell), ignite)
$s_c = \delta_{int}(s_c)$
}
else
lifetime(c) = lifetime(c) - lifetime(Cell)
}

$\mathbf{lifetime}(\mathbf{s})$
return min {lifetime(c) | $c \in Cell$}

Fig. 3. DEVS atomic model of forest fire spread

In this model, we assume there is a direct communication between neighbor cells to exchange the events $ignite$. With a modular modeling, when a cell ignites its neighbors (see Fig. 4), it makes an internal state change to compute the output through the function $\lambda()$, which sends out the event $ignite$ for all neighbors. Then, the execution of $\delta_{ext}()$ of each neighbor puts fire in those cells. In our non-modular modeling, when a cell tries to ignite its neighbors through the autonomous state change, the internal function $\delta_{int}()$ calls to $\delta_{ext}()$ of each neighbor cell to ignite it. Such a communication optimizes the M&S structure and decreases the number of exchanged messages between cells.



(a) time = t $u.t$ (units of time) (b) time = t + α $u.t$

Fig. 4. Spreading of fire in the two dimensional plan using Moore neighborhood.

1) Identification of Ignitable Neighbor Cells: knowing that the wind direction impacts the spread fire and effectively not all neighbor cells are ignited when some cells burn, we develop a specific function to compute the potential neighbor cells which will be ignited. In fact, from aerial view, the spread fire takes the form of an ellipse according to the wind direction. Consequently, a burning cell does not necessarily ignite its eight neighbors necessarily but only some of them for a point of time. We can remark a neighbor situated in the same wind direction will be ignited unavoidably. However, a neighbor in the opposite wind direction will have less chance to be ignited. So, we can admit that the fire spread in neighbor cells follows a sigmoid curve according to the neighbor position and wind velocity (see Fig. 5).

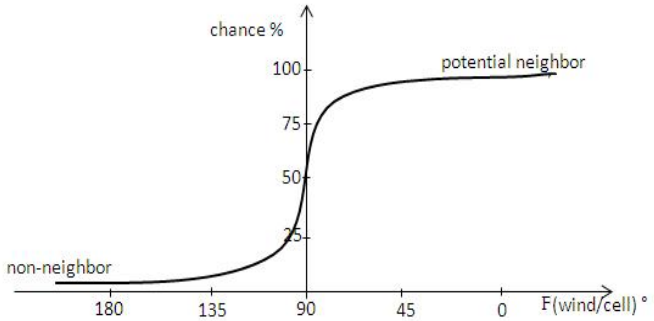


Fig. 5. Neighboring curve according to wind direction and cell position.

2) Forest Mapping: In our approach, the fire is spread on the forest map that the user chooses or defines graphically. This map defines the cell positions and fuels. Each cell has a fuel constant that the user attributes. A zero fuel for a cell means that the cell is nonflammable. However, a non-zero fuel for a cell means that the cell is inflammable; consequently, the corresponding cell behaves according to the DEVS model in Fig. 6.

 ISBN: 978-1-61208-308-7

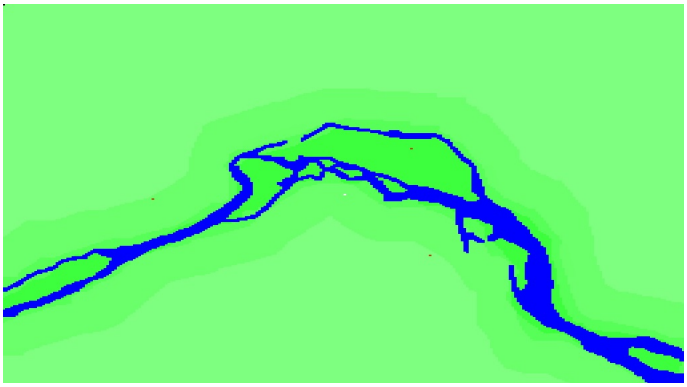


Fig. 6. Forest map with a river.

On this map, the user can identify cells with high risks to compute the reaching time to such cells, using the simulation of the forest fire spread.

C. Simulation of the Proposed Model

In DEVS, the simulation mechanism is well formalized. The structure is made from the model to simulate. So, according to the size of the model to simulate, a computation of the needed heap memory could be estimated. In order to simulate our forest fire spread model, only a basic simulator and a connected root manager are needed to make simulations (see Fig. 7); so, a small size of heap memory is useful to simulate the proposed model.

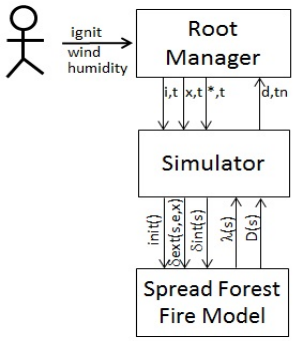


Fig. 7. DEVS simulation structure for non-modular (atomic) model.

The root can interact with user data to update the data of the model to simulate. The user can pause the simulation and then modify any parameter of the model: map data, wind direction and speed or put the fire in inflammable cells. Then, the user re-runs the simulation at the paused time with the new data.

V. Design and Implementation of the Proposed Model

We design the proposed forest fire spread model in the object-oriented paradigm (see Fig. 8). Each cell is designed with a class which holds the corresponding behavior. The different cells are designed into classes `NFlammable` and `Inflammable` to implement the two cell categories non-flammable and inflammable, respectively. The uniform variables $Wind$ and $Humidity$ are sharable among all cell classes.

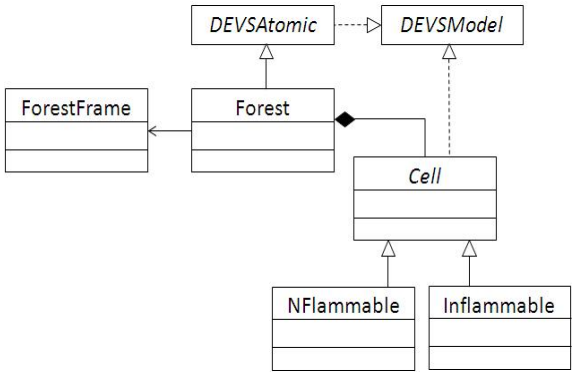


Fig. 8. Class diagram of the forest fire spread model.

This static structure could be replaced by a dynamic one in which we define two lists to hold active and passive cells separately. The active cell list holds cells for which the current state is active and the passive cell list holds cells for which the current state is passive. Thus, a cell migrates from a list to the other according to the cell lifetime. At the end, these lists avoid to visit all cells, i.e., to browse all the cell space to execute the internal state change $\delta_{int}(s)$ and computes the lifetime of current state of the model only from the active cells. Consequently, the run-time of the function $lifetime(s)$ will be reduced.

This design is extensible, it can integrate new cell categories with specific behaviors. The designer extends the class `Cell` and describes the DEVS behavior of the new cell class by implementing the interface `DEVSModel`. The class `Forest` instanciates cell objects according to the saved map description. The `Forest` object is a DEVS atomic model that will interact with the simulator to produce behavior. Note that the structure of this object is static, i.e., cell instances created first will remain until the end of the simulation. In addition, the class `Forest` notifies the graphical user class `ForestFrame` about the state changes occurring in cell objects in order to visualize the spread of fire on the chosen map.

The simulation core jDEVSPattern is developed according to the design proposed in [16] using DEVS simulation algorithms [7] and Java language. Often, the use of standard DEVS simulators certified by the community are well-suitable like DEVSJava, CD++, etc., for which many applications were modeled and simulated with success. However, we privilege this simulator for the following reasons:

1) the design of the simulator is based on software engineering design patterns. So, the software simulation reuses existing, approved and well-known solutions.
2) the validation of the simulator through the simulation of complex systems.

 ISBN: 978-1-61208-308-7

Fig. 9 shows the simulation of fire spread on the chosen map seen on Fig. 6 at the end stage.

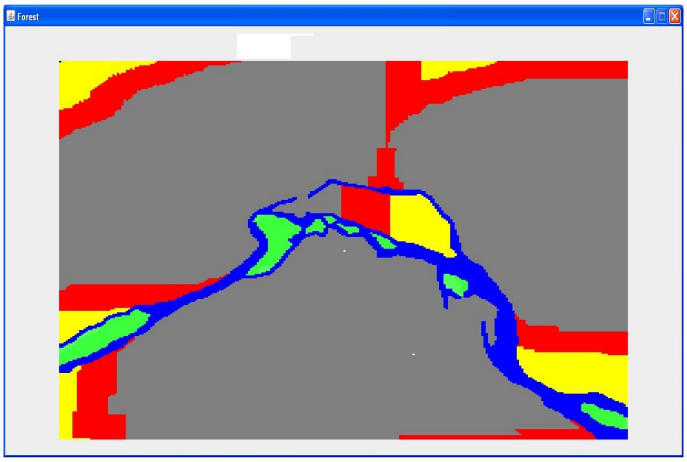


Fig. 9. Simulation of forest fire spread for a specific scenario.

In the above figure, we can note that zones enclosed by the river are still unburned due to the fact that the fire can not cross water area.

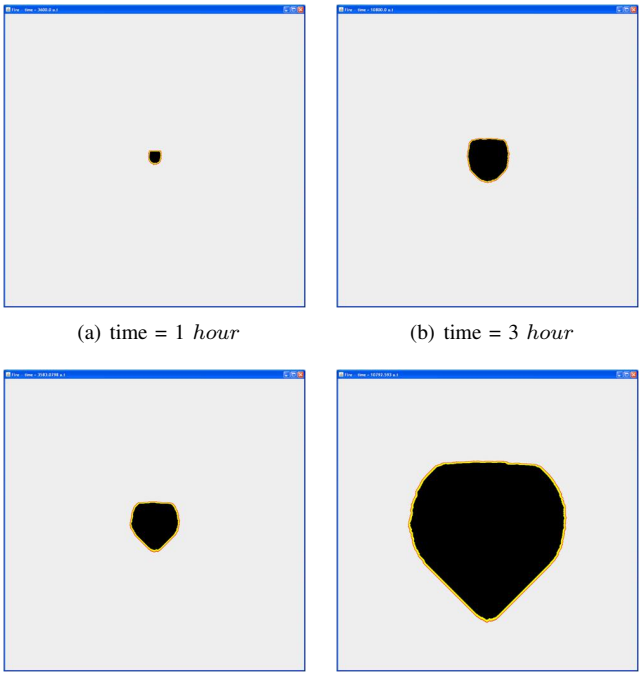
VI. Simulation results and comparative study

A. Simulation of the forest fire spread model

In order to experiment the proposed model for the forest fire spread and analyze the fire evolution, we execute two simulations with the same kind of vegetation and different wind speed. The wind direction is supposed blowing from north to south (0°). The forest square is 25 km^2 designed with cell space 1000x1000 $cells$ and cell size $5m$x$5m$. The simulations are conducted on a personal computer with the following characteristics: Windows XP professional, Intel Core(TM) 2 Duo CPU 3.0 GHz and 1.97 GB of RAM. The lifetime of states burning and ember are estimated from a well-known rule of firemen from the Mediterranean forests. They estimate the ROS as 3% from wind speed. Consequently, for the two expected scenarios in which the wind speeds are 3 km/h and 10 km/h, we deduce the following values of ROS: 0.025 m/s and 0.083 m/s, respectively. Note that for the two scenarios we ignite the cell situated in the middle (cell(500,500)).

In Fig. 10, we show the spread of forest fire at two stages (1 hour and 3 hour) after event $ignite$. We can see clearly in scenario 1 that the fire spreads slowly in case of a calm wind; comparing to the scenario 2 in which the fire spreads quickly causing an important burned area greater than in scenario 1. In addition, the shape of fire in scenario 1 is a circle, lightly flat from north side due to the fact that the wind blocks the fire spread in north direction. In scenario 2, the shape of fire has an ellipse one, with an oriented cone to south. In fact, this cone is driven by the wind direction and its form (pointed or large) depends on the wind speed.

The propagation rules of fire, that we use in the proposed model, are different from Rothermal rules, and developed by



(a) time = 1 $hour$ (b) time = 3 $hour$

(c) time = 1 $hour$ (d) time = 3 $hour$

Fig. 10. Fire spread on GUI `ForestFrame`.

using our own skills. Unfortunately, it is difficult to validate the forest fire spread model due to the lack of real data. An issue, that we will explore, consists in using fuzzy logic to well estimate the ROS according to firemen knowledge, and to make easier the validation process.

B. Performance analysis

To analyze the time execution (second s) and resource allocation (mega-byte MB) of our model, we re-conduct the scenario 1 in which the simulation continues until there is no cell to ignite and all burning cells change state to unburned using different cell spaces 100x100, 200x200, 500x500, 1000x1000 and 2000x2000 $cells$. The cell (0, 0) is ignited initially.

TABLE I
Execution time of non-, partial- and modular models

Exec. Time (s)	100x100	200x200	500x500	1000x1000	2000x2000
non-modular	0.1	0.3	4.2	26.1	194.4
partial-modular	36.2	232.2	288.2	282.8	-
modular	154.6	1088.2	1320.5	1337.6	-

From Table I, we see that simulation of cell space size less than 1000x1000 $cells$, the execution time takes some few milliseconds. The enhanced simulation structure and the optimized model avoid sending out and receiving messages through ports and exchanged between the simulators and the coordinator to ignite the neighbor cells. This fact, leads to a minimum time CPU to execute the fire spread behavior. However, for the cell space size more than 1500x1500 $cells$, the

 ISBN: 978-1-61208-308-7

execution time of the corresponding simulation is important (few seconds).

Comparing the execution times of partial-modular and modular approaches that were carried out on a laptop Toshiba with Intel Celeron 1.6 GHz CPU, 1.2 GB of heap memory and Windows XP operating system using DEVSJava version 3.0 and end simulation at tn = 11 $hours$ (the given results are extracted from [15]), our approach is still very efficient, although our simulations are carried out on personal computer well-equipped (Intel Core(TM) 2 Duo CPU 3.0). In fact, the design of our model, which is based on a non-modular approach and in which cells are modeled with state variables, makes the main difference with the approaches that design cells with atomic models.

The simulation of our model uses a small size of heap memory, which increases essentially according to the size of cell space (cell space with 1000x1000 $cells$ consumes 44 MB). A dynamic structure could be used to design the cell space instead of a static one by keeping temporary in memory only active cells, in order to optimize the computations and to manage efficiently the heap memory.

VII. Conclusion

In this paper, we privileged the non-modular approach to model the forest fire spread using DEVS instead of partial-modular and modular ones; so, we avoid in our modeling the classical rule that consists on designing each cell with an atomic model and we decide to incorporate it as a state variable of the forest fire spread model. Note that this model is designed using the object-oriented paradigm, which allows to design this state variable with an object instance.

The simulation of the proposed model gives correct results by analyzing the simple conducted scenarios. On the other hand, simulation performances (execution time and heap memory) are more advantageous than those given by the modular and partial-modular implementations of DEVS-Fire. In the near future, we will compare our non-modular model with other forest fire spread models and real fire cases in order to validate the given results. The main difficulty will consist in reproducing real fires for which it is difficult to collect accurate data (fuels, landscape, wind direction, and wind speed) [17].

Currently, we are working to introduce fuzzy logic rules of terrain (slope and aspect), weather and vegetation to improve the computation of the ROS parameter which influences mainly the behavior of fire and to solve the problem of accurate data for modeling the input data of the simulation. This will give more significant results and insights to firemen.

References

[1] L. V. Bertalanffy, General System Theory. Dunod edition, 1973.

[2] P. A. Fishwick, Simulation model design and execution: building digital worlds. Prentice hall, 1995.

[3] X. Hu, Y. Sun, and L. Ntaimo, "Devs-fire: Design and application of formal discrete event wildfire spread and suppression models," Simulation: Transactions of the Society for Modeling and Simulation International, vol. 88, no. 3, March 2012, pp. 259–279.

[4] B. P. Zeigler, Theory of Modeling and Simulation. Wiley&Son, 1976.

[5] N. Giambiasi, B. Escudé, and S. Ghosh, "Gdevs: A generalized discrete event specification for accurate modeling of dynamic systems," Simulation: Transactions of the Society for Modeling and Simulation International, vol. 17, no. 3, September 2000, pp. 120–134.

[6] G. Wainer and N. Giambiasi, "Application of the cell-devs paradigm for cell spaces modelling and simulation," Simulation: Transactions of the Society for Modeling and Simulation International, vol. 76, no. 1, January 2001, pp. 22–39.

[7] B. P. Zeigler, H. Praehofer, and T. G. Kim, Theory of Modeling and Simulation. Academic Press, 2000.

[8] F. A. Shiginah and B. P. Zeigler, "A new cell space devs specification: Revewing the parallel devs formalism seeking fast cell space simulations," Simulation Modelling Practice and Theory, vol. 19, no. 5, May 2011, pp. 1267–1279.

[9] R. O. Weber, "Modelling fire spread through fuel beds," Progress in Energy and Combustion Science, vol. 17, no. 1, 1991, pp. 67–82.

[10] L. Hernandez Encinas, S. Hoya White, A. Martin del Rey, and G. Rodriguez Sanchez, "Modelling forest fire spread using hexagonal cellular automata," Applied Mathematical Modelling, vol. 31, no. 6, June 2007, pp. 1213–1227.

[11] A. Muzy, J. J. Nutaro, B. P. Zeigler, and P. Coquillard, "Modeling and simulation of fire spreading through the activity tracking paradigm," Ecological Modelling, vol. 219, no. 1–2, November 2008, pp. 212–225.

[12] B. Nader, J. B. Filippi, and P. A. Bisgambiglia, "An experimental frame for the simulation of forest fire spread," in Proceedings of the 2011 Winter Simulation Conference, December 2011, pp. 1010–1022.

[13] Y. Sun and X. Hu, "Performance measurement of devs dynamic structure on forest fire spread simulation," in Proceedings of the AI, Simulation and Planning in High Autonomy Systems (AIS 2007), February 2007, pp. 75–80.

[14] ——, "Performance measurement of dynamic structure devs for large-scale cellular space models," Simulation: Transactions of the Society for Modeling and Simulation International, vol. 85, no. 5, May 2009, pp. 335–351.

[15] ——, "Partial-modular devs for improving performance of cellular space wildfire spread simulation," in Proceedings of the 2008 Winter Simulation Conference, December 2008, pp. 1038–1046.

[16] M. Hamri and L. Baati, "On using design patterns for devs modeling and simulation tools," in Proceedings of the 2010 Spring Simulation Multiconference-Symposium Theory of Modeling Simulation-DEVS Integrative MS symposium, April 2010, doi:10.1145/1878537.1878664.

[17] F. Gu, X. Hu, and L. Ntaimo, "Towards validation of devs-fire wildfire simulation model," in Proceedings of the 2008 Spring simulation multiconference, April 2008, pp. 355–361.

 ISBN: 978-1-61208-308-7

ComCAS: A Compiled Cycle Accurate Simulation for Hardware Architecture

Adrien Bullich, Mikaël Briday, Jean-Luc Béchennec and Yvon Trinquet
IRCCyN - UMR CNRS 6597
Nantes, France
Email: {first name.last name}@irccyn.ec-nantes.fr

***Abstract*—This article is in the context of real-time embedded systems domain. These critical systems require an important effort in validation and verification that can be done at many abstraction levels, from high-level application model to the actual binary code using an accurate model of the processor. As the development of a handwritten simulator of a processor at a cycle accurate level is a difficult and tedious work, we use HARMLESS, a hardware description language that can generate both a functional and a cycle accurate simulators. The latter gives a temporal information of the simulation execution, but at the cost of a heavy computation overhead. This paper applies the compiled simulation principles to a cycle accurate simulator. It shows that this simulation mechanism can reduce computation time up to 45%, preserving timing information.**

Keywords*—*Cycle Accurate Simulation; interpreted simulation; compiled simulation; HADL

I. INTRODUCTION

Verification of a real-time application is a huge and difficult problem. It must be led throughout the development cycle on functional and extra-functional aspects (temporal aspects, safety, etc.). Our work takes place in the last stages of the development process, when the actual binary code of the application is available, just before the final test on the real target.

A simulation scheme should be chosen according to the studied field, pursued objectives, and the abstraction level required. The lower is the abstraction level, the higher is the simulator complexity. For hardware simulation, the implementation is complex, time consuming, and errors are difficult to avoid. To alleviate this complexity, a Hardware Architecture Description Language (HADL) may be used. With such a language, the complexity remains partly hidden. For the work presented herein, HARMLESS [1] has been used to build the simulators. The HARMLESS compiler can generate both functional simulators, *i.e.*, Instruction Set Simulator (ISS), and temporal simulators, *i.e.*, Cycle Accurate Simulator (CAS) from a common description. We consider here especially CAS, as timings of the application should be taken into account for real-time systems.

A CAS simulator requires much more computation time than an ISS. In [1], the CAS is about 7 times slower than the functional one on a simple PowerPC processor with a 5-stages pipeline. So, CAS related computation has room for improvement. Here, we propose to explore a technique to improve the speed of CAS, which is a compiled simulation for CAS.

The paper is organized as follows: Section II presents the related works; Sections III and IV explain the current model of interpreted simulation and the new approach using the compiled simulation; Section V evaluates the model size and Section VI presents some results on a set of benchmarks; Section VII concludes this paper.

II. RELATED WORKS

Many HADLs have been proposed in the literature. Some of these HADLs only focus on the functional aspects of the instruction set they describe. So, the associated toolset is only able to generate an ISS. nML [2] and ISDL [3] are examples of this kind of HADLs. Other HADLs add a micro-architecture description from which a temporal behavior is constructed. For instance, LISA [4], MADL [5] and HARMLESS [6], [7] have the ability to generate an ISS and a CAS.

An interpreted simulator simulates the execution of a binary executable by doing the same steps as the hardware it simulates. So, for each binary instruction an ISS does the following steps: instruction fetch, instruction decode, and instruction execution. In addition, a CAS computes instructions dependencies, controls concurrent accesses to the buses, register files, and generally any computing resource of the architecture.

A compiled simulator is customized to execute a particular binary executable. Knowing the binary executable at the compilation stage, it allows to remove from the execution stage all the tasks that depend on the executed instruction only. As a result, a compiled simulator exhibits better performance than an interpreted one, but it has a longer compilation time. Since compilation is done less times than execution, classically one compilation for several executions, a compiled simulator offers a global gain of time. However, a compiled simulator is less flexible because it is attached to a particular program: if the user wishes to simulate another program, he needs to compile again.

For an ISS, compiled simulation consists in Binary Translation (BT) ([8] or [9]). First, the binary executable one wants to simulate is translated to a native binary of the host simulation platform. Then, the native binary is executed on the host simulation platform.

For a CAS, few methods exist for compiled simulator generation. The technique of BT cannot easily be adapted to CAS, but solutions exist [10], coupling interpreted parts and translated parts. We also find statistic approaches, called Cycle Approximate Simulator, based on the sampling of instructions [11]. However, it is not exactly equivalent to a CAS, because of errors margin.

To the best of our knowledge, the technique of compiled simulation has not yet been employed to speed CAS up, because of the restrictions it implies: it is difficult to determine statically the evolution of the micro-architecture. However, this is the main contribution of the paper.

III. INTERPRETED SIMULATION MODEL

The contribution of the compiled simulation must be assessed in comparison with the associated interpreted approach. In this section, we present the interpreted model of the Cycle Accurate HARMLESS-based simulator [1], that is the base of our ComCAS model.

 ISBN: 978-1-61208-308-7

In the interpreted model, instructions of the application code are decoded and executed during the simulation. The model of a cycle accurate processor includes the instruction set and the memory model for functional execution and all the micro-architecture related parts that alter timings, as presented in the development chain in Figure 1.

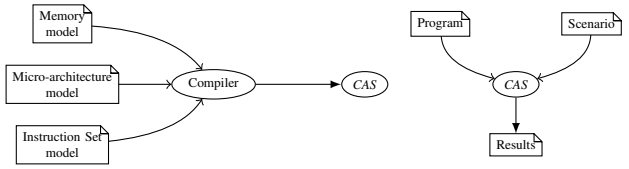


Fig. 1. The development of a CAS requires the modeling of the instruction set, the memory and the micro-architecture

One important micro-architecture unit is the processor pipeline: it has an influence on timings of the processor and this is the most expensive in computation time. Ideally, each instruction in each pipeline stage progresses to the next stage at each processor cycle. Actually, an instruction can be blocked in a pipeline stage because of *hazards*. Hazards are classified into three categories [12]: *structural hazards* are the result of a lack of hardware resources; *data hazards* are caused by data dependencies between instructions (for example between stages W and D in Figure 2); and *control hazards*, which occur when a branch is taken in the program (one or more instructions that just follow the branch according to the branch delay that should be flushed). When a hazard is encountered, it is solved by stopping a part or all parts of the pipeline. This is called a *pipeline stall*.

Sequential pipelines are considered in this paper (*i.e.*, there are neither pipelines working in parallel, nor forking pipelines). The pipeline behavior is modeled in HARMLESS using an automaton, where a state represents the pipeline state at a particular time (see Figure 2).

In [1], the authors use the model of finite automata, because the system can be considered as a discrete transition system, a transition being taken at each cycle, as in Figure 2. The

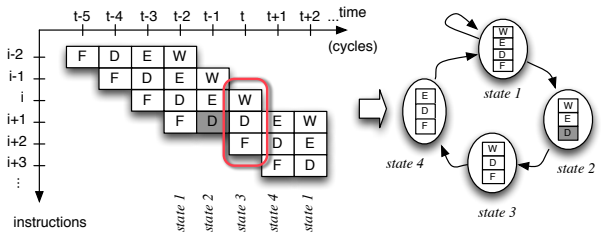


Fig. 2. A state of the automaton represents the state of the pipeline at a given time. Here, the pipeline has 4 stages. F: instruction is fetched, D: instruction is decoded and registers are read, E: instruction is executed, and W: the result is written into a register.

contribution of this paper is based on this definition. A state represents the system in a particular cycle. A state is defined by:

- which instruction is in each stage of the pipeline;
- the state of internal resources.

Internal resources are elements of the micro-architecture that are used only by the pipeline. Their availability allows or not the progression of an instruction in the pipeline. Stages of the pipeline themselves are considered as internal resources.

Instructions that use the same resources in the same pipeline stage are grouped together to form *instruction classes*. This is the case for instance for arithmetic instructions that read two registers, make a calculation, and write the result into a third register. Since internal resources depend only on the instruction class and on the pipeline stage, they are not needed at run time.

As a result, a transition represents a discrete event that brings the system from a state to another. It is determined by the state of *external resources* and the next instruction class that enters the first stage of the pipeline.

External resources are elements that are not used only by the pipeline, *i.e.*, their state is defined in other micro-architecture parts such as memory caches. The availability of these external resources has an influence on the evolution of instructions in the pipeline, too. Moreover, as they are external of the pipeline model, their availability is determined during the execution.

The content of states is abstracted, and information required for the simulation is gathered on transitions. For this reason, transitions are labeled with *notifications* (signaling if a particular event happens or not).

We can now formalize the model. Let AI be an automaton defined by $\{S, s_0, ER, IC, N, T\}$, where:

- S is the set of states;
- s_0 is the initial state (empty pipeline) in S;
- ER is the first alphabet of actions (external resources);
- IC is the second alphabet of actions (instruction classes);
- N is the alphabet of labels (notifications);
- T is the transition function in $S \times ER \times IC \times N \times S$.

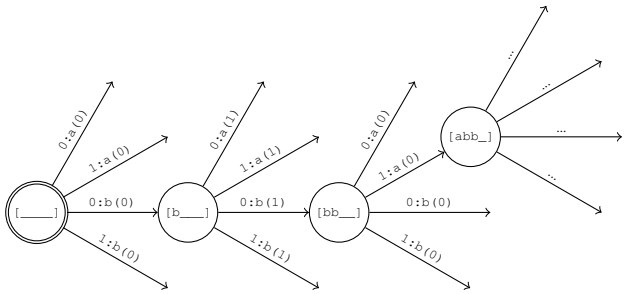


Fig. 3. Automaton in interpreted simulation: `0:b(1)` means that the external resource is free (`0`), that the instruction `b` may enter the pipeline and that the notification happens (`1`)

In the example of Figure 3, the notation `[b_a_]` represents the state of the 4-stages pipeline: it means that instruction of class `b` is in the first stage and instruction of class `a` is in the third stage. There are no other instruction classes for readability. We have only one notification that represents the *entry of an instruction in the second stage of the pipeline*. There is one external resource. The instruction class `b` needs to take the external resource to enter the pipeline.

During the simulation, both the state of external resources and the instruction class of the next instruction that will enter the pipeline are required to determine the next state of the automaton. When a transition is taken, notifications related to the transition are given to the simulation engine to interact with other micro-architectural parts.

 ISBN: 978-1-61208-308-7

IV. ComCAS model

In this section, we adapt the interpreted model to be a compiled one: the ComCAS model.

The compiled simulation differs from the interpreted simulation in the repartition of tasks between compilation and execution. We recall that, in our case, the task is the analysis of the program. An interpreted simulator analyzes the program during the execution. A compiled simulator analyzes the program during the compilation.

Because of this change, the compiled simulation has a faster execution than the interpreted simulation. However, the compiled simulation has a longer compilation time than the interpreted simulation. This is not necessarily a problem: usually, the compilation is done only once, while the execution is performed several times.

A compiled simulation is run for a special architecture and for a special program. Consequently, the simulator is less flexible, attached to a particular program. If the need is to simulate several programs, the interpreted simulation will be more efficient. But, if the need is to simulate only one program with different scenarios, the compiled simulation will be more efficient.

In Figure 1 and Figure 4, we can see the difference between the development chain of the interpreted simulation and the compiled simulation, respectively. We notice especially that for the interpreted simulation the program is at the end of the development chain and at the beginning for the compiled simulation.

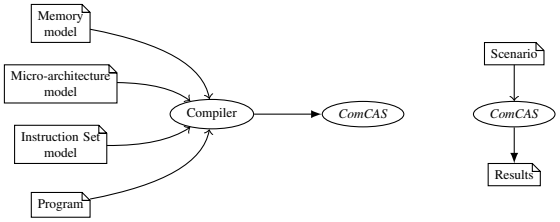


Fig. 4. The development of a compiled CAS requires to move the program analysis in the compilation

To transform the interpreted model into a compiled model, we need to add some information about the program. We first need its memory mapping, *i.e.*, the location of each instruction in memory, the corresponding Program Counter (PC) and the stack of called function (a stack of PC, in order to return to previous functions). Then, the determination of our system is given by:

- which instruction is in each stage of the pipeline;
- the state of internal resources;
- the position in the program (the Program Counter);
- and the stack of called functions.

With this model, instructions become labels on the automaton and no more actions are needed. Indeed, we only determine the evolution of the system with external resources and instructions become an information we get out of this run. However, it cannot be reduced to a simple notification (a boolean information), so we add the PC on the transition label.

We can formalize our ComCAS model as it follows.

Let AC be an automaton defined by $\{S, s_0, ER, I, N, T\}$, where:

- S is the set of states;
- s_0 is the initial state (empty pipeline, initial PC, empty stack) in S;
- ER is the alphabet of actions (external resources);
- I is an alphabet of labels (instructions);
- N is an other alphabet of labels (notifications);
- T is the transition function in $S \times ER \times I \times N \times S$.

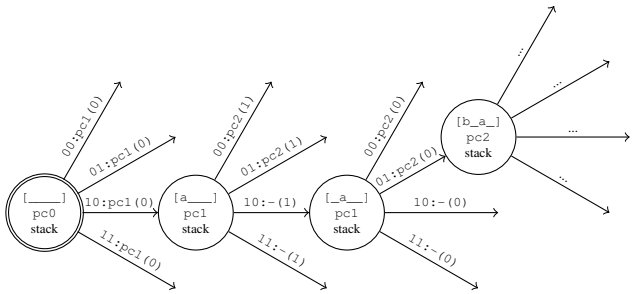


Fig. 5. Automaton in compiled simulation: `10:pc1(0)` means that the first external resource is free and the second taken (`10`), that the instruction with PC `pc1` enters the pipeline and that the notification does not happen (`0`)

In the example from Figure 5, we have only one notification that represents the entry of an instruction in the second stage of the pipeline. There are two external resources. The instruction `b`, with PC `pc2`, needs to take the second external resource to enter the pipeline.

The management of branches uses a specific external resource. If this resource is taken, the branch is taken and conversely. The use of an external resource is mandatory because in the general case, the branch target can only be computed at runtime. During the simulation, we can detect if a branch is taken and define dynamically the value of this resource. In order to represent the latency of the branching, according to the branching policy, another specific external resource could be employed to model *control hazards*. If the micro-architecture uses a branch predictor, the simulator would emulate this branch predictor and define dynamically the value of the corresponding external resource. While the resource is defined to be taken, the instruction that follows could not enter the pipeline.

An example is given in Figure 6. The first external resource represents the branch management (used in this case for the branch `b` to `pc3`). If it is taken, then the model goes to the target PC (`pc3`), else it goes to the next PC (`pc1`). The second resource represents the branching latency. As long as it is taken, no instruction can enter the pipeline.

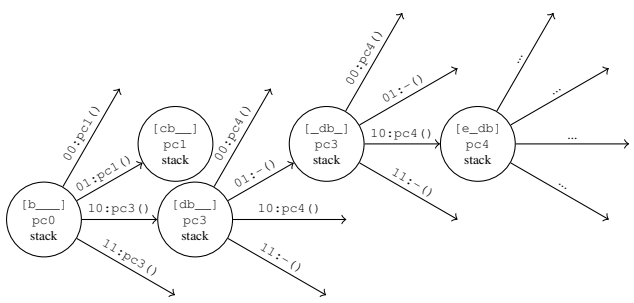


Fig. 6. The second external resource specifies if a branch (like `b`) is taken or not. The first external resource is used to model branching latency (delaying in this case the entry of instruction `e` in the pipeline).

 ISBN: 978-1-61208-308-7

The compiled simulation needs to compute statically the control flow of the program, in order to model it during the compilation. In the case of indirect branches or code self-modifying, this computation is impossible unless we execute the different scenarios of the program. This is the main restriction of our model. Without the possibility to determine statically the target of indirect branches, the solution is to plan the different possibilities. Unfortunately, it leads to a considerable increase of the automaton's size.

Indirect branches are unavoidable if we want to model functions calls, because of `RETURN` instructions. This specific problem is solved with a PC stack that is added in states. The stack in states allows to memorize original PC when a `CALL` instruction is executed. When a `RETURN` instruction is executed, this stack allows to determine the target PC of the branch, during the compilation.

The process is the following: when a `CALL` instruction enters the pipeline, we push the next PC onto the stack, and we branch to the target PC. When a `RETURN` instruction enters the pipeline, we pop a PC from the stack, and we branch on it. An example is given in Figure 7.

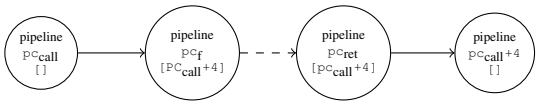


Fig. 7. A `CALL` function pushes the original PC onto the stack, and a `RETURN` function pops the PC from the stack

The main advantage of the compiled simulation is to move a computation part from the runtime to the compile time. This is the case for *data hazards* that are handled in the interpreted simulator using an external resource. This is a costly task that could be solved at compile time in ComCAS: the instructions in the pipeline are known, so we can determine all the registers that are read and written statically.

V. Valuation of the number of states

In order to give an idea of the complexity of ComCAS model, we propose to evaluate the number of states.

A state is composed of the pipeline state, the PC corresponding to the last instruction read and the stack of called functions. In a first step, we will consider there is no stack in states, and we will add this feature afterwards. The global method consists in counting pipeline states for a given PC. We find three different situations in the control flow: linear configuration, beginning of the program and branching configuration.

In a linear configuration, for a given PC, one past exists. Consequently, the state of the pipeline is only determined by pipeline stalls. The problem is reduced to a combinatory one: if s is the number of stages, we count C_s^k (k among s) possible pipeline states with k instructions inside ($k \in [0; s]$). Thus, the total number of pipeline states is $\sum_{k=0}^{s} C_s^k$.

If the PC points at the beginning of the program, pc_n with $n < s$, it is impossible to put more than n instructions in the pipeline. So, in this case, the previous value is truncated to $\sum_{k=0}^{n} C_s^k$. To simplify computation, from now on, we use $f_s : n \to \sum_{k=0}^{n} C_s^k$. And we know that $f_s(s) = 2^s$.

At this step of our computation, we can valuate the number of states in a perfect linear program (with no branch). Let i be the number of instructions. The first s instructions are in the second case (beginning of the program), and others $i - s$ instructions are in the first case (endless linear configuration). It gives: $\sum_{k=0}^{s-1} f_s(k) + (i - s).2^s$.

This value is a maximum, and it is reached if every pipeline states is explored. It is the case when an external resource manages the entry of instructions in the first stage (bus access or cache miss), allowing all stalls arrangements.

The number of pipeline states is larger if we include branches in the control flow. Let us consider the case in Figure 8, with $k < s$. In this situation, if we put j instructions in the pipeline with $j \leq k$, the branch is not visible in the pipeline. Thus, we remain in the same previous situation: C_s^j pipeline states. But, if we put j instructions in the pipeline with $j > k$, then for each pipeline stalls arrangement two pipeline states exist, with two different pasts. So, we count $2.C_s^j$ pipeline states. The total number of pipeline states is $\sum_{j=0}^{k} C_s^j + \sum_{j=k+1}^{s} 2.C_s^j$ It is equivalent with $2^{s+1} - f_s(k)$.

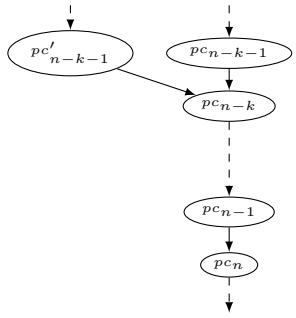


Fig. 8. A branch configuration in the control flow. If $k < s$ then in PC pc_n the pipeline can remember two pasts.

For one branch situation, we have k varying in $[0; s - 1]$. Let b be the number of branch targets. So, the total number of states becomes:

$$\sum_{k=0}^{s-1} f_s(k) + (i - s - s.b).2^s + b.(s.2^{s+1} - \sum_{k=0}^{s-1} f_s(k)) \quad (1)$$

It is equivalent to:

$$(1 - b).\sum_{k=0}^{s-1} f_s(k) + (i - (1 - b).s).2^s \quad (2)$$

The expression is valid if branch configuration is the same as the one we give in Figure 8. It means that two conditions arrise:

- branch targets are separated by more than s instructions;
- no branch is less than s instructions after a branch target.

In fact, we can confirm that the first condition does not degrade our valuation. The second condition is more important and precludes too small loops.

The analysis of our valuation reveals that the number of states is linear with the number of instructions, and exponential with the number of stages. We can compare the expression with the number of states in interpreted simulation: $(ic + 1)^s$,

which is more exponential considering s, ic being the number of instruction classes.

In our valuation, we have not yet considered the use of PC stack in states. We can see in Figure 9 the effect on the control flow of the add of PC stacks.

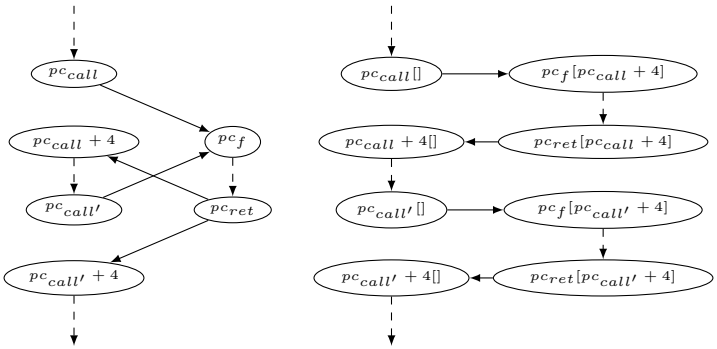


Fig. 9. Stack consideration consists in an inline in the control flow. These two automata are equivalent.

It can be regarded as inlining: functions' code is duplicated. If we consider this new control flow, our reasoning is the same. Let us call i' the number of couple of instruction and PC stack, and b' the new number of branch targets. Thus, the number of states simply becomes:

$$(1-b').\sum_{k=0}^{s-1} f_s(k) + (i' - (1-b').s).2^s \qquad (3)$$

The influence of this inlining is very dependent on the code (see Table I). For example, we observe that programs using software floating point numbers increase significantly the size of the automaton, and make difficult the construction of the model.

VI. Tests and Performance

In this section, we present experimental results about performance of ComCAS model in comparison with the interpreted simulation.

The architecture simulated in these tests is similar to a PowerPC 5516 from Freescale, with a *e200z1* core. The pipeline has been resized from 4 to 5 to increase the size of the model. We ran the benchmarks of Mälardalen [13]. Simulations are made with an Intel *Core i7@3,4Ghz* computer. We execute 50 000 times each program.

We give in Table I an illustration of the influence of the inlining and the number of states, obtained by ComCAS tool. This allows to confirm that if a function is called once during the execution, PC stack has no influence on the size of the model. We note that the number of states is smaller than the valuation we can compute, because the model does not explore every pipeline states. With particular external resources (making every pipeline states possible) we get the same result than our valuation. To allow a comparison, with the same configuration the interpreted model gets 1 024 states. Smaller is the code, smaller is our model.

Figure 10 represents the performance of ComCAS model in comparison with the interpreted method for the execution time. In the compiled approach, the generation of the simulator is more complex, as it requires to generate the ISS, analyze the program (using the ISS) and build the simulator. This time consuming compilation step is largely counterbalanced by a faster execution time, which is done several times.

TABLE I. Influence of the inlining: **i** is the number of instructions, **b** the number of branch targets, **i'** the number of instructions with the inlining and **b'** the number of branch targets with the inlining

Program	i	b	i'	b'	States
adpcm	2 243	79	3 308	79	75 588
bs	84	4	84	4	2 061
compress	867	40	1 027	43	24 586
cover	145	7	145	7	3 434
crc	322	11	584	19	13 022
duff	88	3	88	3	2 101
expint	185	8	185	8	4 544
fdct	692	3	692	3	14 638
fibcall	58	3	58	3	1 447
fir	144	5	144	5	3 398
insertsort	131	3	131	3	2 961
janne_complex	76	6	76	6	1 974
jfdctint	551	4	551	4	11 605
lcdnum	74	4	74	4	1 768
matmult	203	7	274	8	6 844
ndes	1 009	31	1 377	47	32 976
ns	116	8	116	8	2 907
nsichneu	12 511	626	12 511	626	275 322
prime	147	8	268	9	6 706

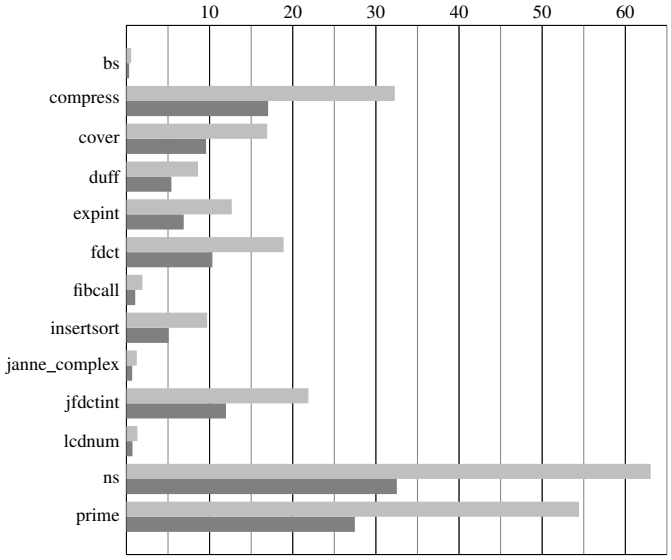


Fig. 10. Comparison of execution time in seconds for 50 000 executions. Gray is for interpreted simulation, and black is for compiled simulation.

The main impact of our model comes from the ability to manage analysis tasks when compiling. In particular, the treatment of the data dependency control in the compilation phase has been implemented in ComCAS. It reduces the execution time by 45,1% on average as we can see on Figure 10, and up to 49,5% with *prime* benchmark. This significant benefit shows the interest for the compiled simulation for the validation of real-time embedded systems.

VII. Conclusion

In this paper, we have discussed the different techniques to implement high speed Cycle Accurate Simulator. We have

 ISBN: 978-1-61208-308-7

developed a model to adapt the compiled simulation approach to Cycle Accurate Simulator and implement it in the ComCAS tool. We have studied the maximum theoretical size of our model and compared performance of our model with the associated interpreted method. These results show that the computation time is reduced by 45% in comparison with the interpreted simulator.

Compiled simulation is efficient because it allows to remove some analysis tasks from the execution step. Even if this technique does not currently handle indirect branches, function calls are taken into consideration to simulate a major part of embedded systems programs.

Future work aims at improving the efficiency of the ComCAS model by using *macro-instructions*. A macro-instruction gathers the behavior and the timing of a set of successive instructions provided there is no external resource used by these instructions. However, an external resource attached to the *fetch* stage is needed and precludes the construction of macro-instructions. The solution could be to take the cache behavior into account to remove this external resource. With this improvement, the size of the automaton would be reduced and the performance of the simulator would be increased.

Another path of improvement would be to use the ComCAS model in a Just In Time simulator. In this case, the interpreted simulator would reduce the automaton *on the fly* when a loop is encountered and would switch its execution to the reduced automaton to improve performance dynamically. This could bring a solution for the problem of indirect branches.

REFERENCES

[1] R. Kassem, M. Briday, J.-L. Béchennec, G. Savaton, and Y. Trinquet, "HARMLESS, a hardware architecture description language dedicated to real-time embedded system simulation," Journal of Systems Architecture - doi: http://dx.doi.org/10.1016/j.sysarc.2012.05.001 [retrieved: august, 2013], September 2011, pp. 318–337.

[2] A. Fauth, J. Van Praet, and M. Freericks, "Describing instruction set processors using nml," EDTC'95: Proceedings of the 1995 European Conference on Design and Test, March 1995, pp. 503–507.

[3] G. Hadjiyiannis, S. Hanono, and S. Devadas, "Isdl: an instruction set description language for retargetability," DAC'97: Proceedings of the 34th annual conference on Design automation, 1997, pp. 299–302.

[4] S. Pees, A. Hoffmann, V. Zivojnovic, and H. Meyr, "Lisa - machine description language for cycle-accurate models of programmable dsp architectures," DAC'99: Proceedings of the 36th ACM/IEEE conference on design automation, 1999, pp. 933–938.

[5] W. Qin, S. Rajagopalan, and S. Malik, "A formal concurrency model based architecture description language for synthesis of software development tools," in Conference on Languages, Compilers, and Tools for Embedded Systems (LCTES'04), 2004, pp. 47–56.

[6] R. Kassem, M. Briday, J.-L. Béchennec, G. Savaton, and Y. Trinquet, "Simulator generation using an automaton based pipeline model for timing analysis," in International Multiconference on Computer Science and Information Technology (IMCSIT'08), Wisla, Poland, October 2008, pp. 657–664.

[7] R. Kassem, M. Briday, J.-L. Béchennec, Y. Trinquet, and G. Savaton, "Instruction set simulator generation using HARMLESS, a new hardware architecture description language," Simutools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques, 2009, pp. 24:1–24:9.

[8] C. Cifuentes and V. Malhotra, "Binary translation: Static, dynamic, retargetable?" Proceedings International Conference on Software Maintenance, 1996, pp. 340–349.

[9] F. Bellard, "Qemu, a fast and portable dynamic translator," Translator, vol. 394, 2005, pp. 41–46. [Online]. Available: http://www.usenix.org/event/usenix05/tech/freenix/full_papers/bellard/bellard_html/[retrieved:august,2013]

[10] D. Jones and N. Topham, "High speed cpu simulation using ltu dynamic binary translation," Proceedings of the 4th International Conference on High Performance and Embedded Architectures and Compilers, January 2009, pp. 50–64.

[11] R. E. Wunderlich, T. F. Wenisch, B. Falsafi, and J. C. Hoe, "Smarts: Accelerating microarchitecture simulation via rigorous statistical sampling," Proceedings of the 30th annual international symposium on Computer architecture, June 2003, pp. 84–95.

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

[13] J. Gustafsson, A. Betts, A. Ermedahl, and B. Lisper, "The Mälardalen WCET benchmarks – past, present and future," in WCET2010, B. Lisper, Ed. Brussels, Belgium: OCG, July 2010, pp. 137–147.

 ISBN: 978-1-61208-308-7

Evaluating Options of Viennese Commuters to Use Sustainable Transport Modes

Gerda Hartl, Gabriel Wurzer
Institute of Architectural Sciences: Digital Architecture and Planning
Vienna University of Technology
Treitlstraße 3, 1st floor, 1040 Vienna, Austria
hartl@iemar.tuwien.ac.at, wurzer@iemar.tuwien.ac.at

***Abstract*—The 2-degree guardrail of global warming, together with accelerating urbanization and growing scarcity of oil, demands a redesign of todays' sprawling cities in order to limit greenhouse gases and bring about efficiently built-up structures. In 2001, 42 percent of commuter paths within Vienna are traveled by private cars. This week-daily, recurring traffic pattern causes tons of Co2 and supports low-density housing at urban fringes. In front of this background, we employ an agent-based simulation model to evaluate if commuters in Vienna who currently use motorized modes (especially car-drivers) have options to change to low-carbon transport modes (pedestrian, bicyclist, public transport) without raising costs of travel time. Using a detailed network, the identified present and alternative routes can be displayed as edge-, node-, or zonal through-traffic, highlighting differences in transport mode usage throughout the city.**

Keywords-commuter traffic simulation; multi modal transportation network; sustainable city; agent-based modeling

I. INTRODUCTION

Economic life in cities today stipulates employees commuting to their workplace (and back again) at rush hours. A commuter is a person who doesn't work at his/her residence, but rather leaves it for their workplace, adopting a week-daily spatial-temporal rhythm. The traffic load caused by these daily paths, irrespective of transport mode, is called commuter traffic. With regard to sustainability aims, bicyclist traffic and pedestrian circulation are transport modes of zero carbon emission. Public transports' usage of fossil fuel causes greenhouse gases, yet compensates these by high rates of passenger occupancy, less overall space for infrastructure and hierarchical service line organization. On the contrary, automobile transports' low occupancy rates cause extensive energy consumption, high output of toxic emissions [1] and long-term effects on land-use allocation [2], i.e., urban sprawl. For example, workplace locations tend to agglomerate while residential locations tend to spread out [3]. Vienna clearly exhibits this pattern, which one of our spatial analysis, based on finely grained statistical data (2001: Statistics Austria, retrieved: July, 2013) has shown. Thus, todays' patterns of private car commuting are a mirror image of cheap oil availability and low restrictions to built-up densities in past urban planning decisions.

If the 2-degree guardrail of global warming, agreed upon in Cancun 2010 [4], is to be taken seriously, a re-design of today's' cities, facing accelerating urbanization until 2050, is crucial. This re-design has to involve both re-densification and reduction of traffic-related emissions, because land use and transportation are a closely intertwined system [5]. In this paper, however, we focus on evaluating status-quo commuters' possibilities to change to "greener" transport modes than they currently use. The reason is that quicker adaptation to rising oil prizes [6] can be expected in the domain of transport mode choice [3].

First, we introduce the utilized data and explain the motivation of our agent-based simulation model. Next, we describe the details of our shortest-path algorithm and compare it to existing literature. Finally, we will discuss benefits and weak points, unsolved issues and future work.

II. MODEL DESCRIPTION

A. Data Description

In a preliminary study [unpublished], district-wise census data (from 2009: Statistics Austria) on commuters who live and work in Vienna, distinguished by the transport modes pedestrian, bicycle, public transport and private car (2001: Statistics Austria, retrieved: July, 2013), were used to build a commuter model. Holding modal split equal, commuter relations were spread to the level of 281 sub-districts, using the weights of employee and workplace distribution, both taken from the Viennese Transport Model of the City of Vienna [7]. Furthermore, this model provides separate GIS transport networks (pedestrian, bicycle, public transport, individual transport), each consisting of nodes and directed edges. In these, maximum speed limits and metric lengths are attributed to the edges (Figure 1).

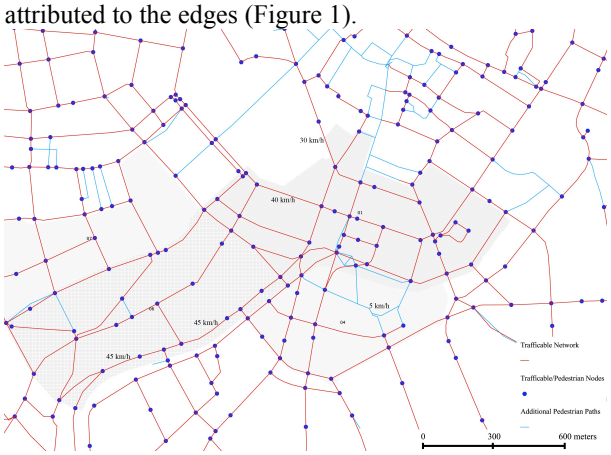


Figure 1. Networks of pedestrian circulation and individual transportation, for the latter examples of maximum speed are displayed.

 ISBN: 978-1-61208-308-7

B. Scientific Background

This information enables us to calculate basic travel time necessary for passing an edge, (travel time = metric length / maximum speed), while running the simulation. We use this indicator as weight during path finding. This way, we can highlight status quos' advantageous usage of fast lanes and highways in individual traffic and research if changes from, e.g., this mode to pedestrian circulation bring about either travel time rises or travel time savings for a single commuter traveling from one zone to another. Travel time is an important indicator in transportation science: being a function of speed and distance, fast transport modes have enlarged travel distance distributions to big amounts because time used for mobility is, on average, almost constant since years [8]. Thus, rising travel time durations for commuting, due to transport mode shifts, are unlikely to be accepted unless hard constraints in the form of monetary costs or regulations are employed. Vienna has recently introduced parking bans for non-Viennese residents, prohibiting surface parking for incoming commuters. Since then, regional trains are on overload. For Viennese residents, parking management has been established in many districts too, partly forcing commuters to switch to other modes. However, our focal point is a precedent one: Can commuters' switchover to low-carbon transport modes be advantageous in terms of time-savings?

C. Model Design

The progression of our simulation model is as follows: in the first step, commuters are distributed to random vertices within their residential zones (source zones). These vertices need to belong to the networks initially required by the commuters, e.g., employees singularly using their private car are distributed to vertices of the individual transport network. Likewise, target vertices are selected in their workplace zones (destination zones). Using travel time on the respective network as weight for procession along the graph, the commuter now determines their initial shortest path, the total duration of which is stored. After all commuters have done so, we have our baseline model of shortest paths for the status quo situation in Vienna. Now, in our reallocation model, commuters try to optimize their baseline paths with regard to travel time by changing to alternative transport modes. Table 1 shows, which alternative modes are allowed for consideration for a current mode, indicating that switches to sustainable transport are preferable.

TABLE I. OPTIONS FOR SWITCHING CURRENT MODE OF TRANSPORT

Current Transport Mode	Allowed Next Transport Mode
Individual Transport	Public Transport, Pedestrian
Public Transport	Pedestrian
Pedestrian	Pedestrian

Concerning transport mode availability, this decision table is strictly logic: once a commuter has, e.g., abandoned his car to use public transport instead, he cannot use it anymore in his subsequent path [9]. Note that we have left out bicycle traffic in this study because its network is largely identical to the pedestrian network, except for higher velocities at the edges.

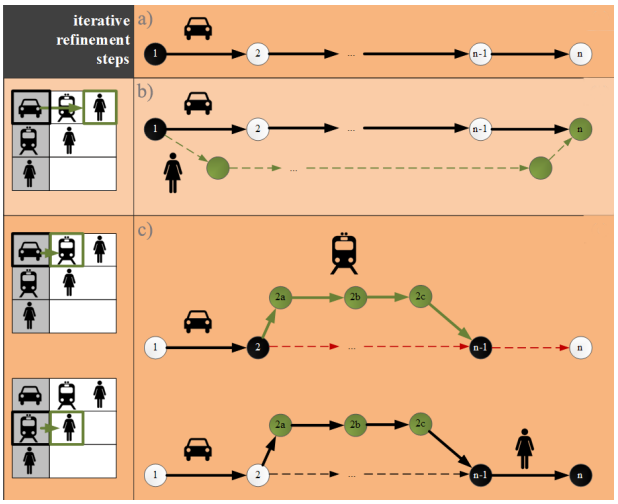


Figure 2. Decision process of a commuter in the reallocation model as based on Table 1.

Figure 2 shows a representation of the agents' decision process, evaluating their options of changing transport modes. Originally, the agent uses his private car (Figure 2a). First, they try to get to their destination by foot (Figure 2b), which does not result in travel time-savings. Consulting Table 1 for other allowed options (here: public transport), which are available at the 2nd node of their initial journey, they try again (Figure 2c). Taking this mode brings them to another current node. Looking up which other modes are available then, pedestrian mode is used to access their final destination node.

D. Model Execution

We have imported the GIS-network data, describe above, into a NETLOGO 3D model [10], extended by a plug-in for shortest path inquiry. Arrival data is loaded from spreadsheets obtained in the commuter model. Due to lack of data, arrival times could not be considered, thus, our simulation pictures rush hour traffic as an "interesting" time span. In detail, the simulation executes the following steps:

- For each agent using a current mode, there are allowed next modes.
- The agent evaluates mode switch options at a current node. It examines if there are alternative routes with the allowed next modes in the following manner: if there is a route with an alternative transport mode which meets the baseline route again, the agent changes his transport mode and takes this route (Figure 2). If the last current node is not the destination node, the agent iterates this process: it examines Table 1 again and takes one of the allowed next transportation means in order to arrive at his final node.
- Once the agent is at his destination node, the travel costs of the alternative route are compared to the

 ISBN: 978-1-61208-308-7

baseline route. In case there is a benefit, the alternative solution is accepted. In all other cases, the agent backtracks to the node at which the disadvantageous fork was conducted. It continues to the next node along the hitherto existing route and tries to switch transport modes again.

III. DISCUSSION

Reviewing the preliminary outcomes of our promising work-in-progress research, there are the following issues, which deserve second thought:

- The given GIS-networks do not reflect reality in the minutest detail. Agents of our model may only change their transport mode, where the different networks (pedestrian, bicycle, public transport, individual transport) explicitly meet at nodes. The outcome is that less mode shifts are performed. The introduction of catchment areas would be useful to enable smoother transfers if, e.g., car-driving agents are in the surroundings of, e.g., public transport stops. This would facilitate better results towards mode shift options, while the networks themselves would not need to be extended.
- Additional information, like parking space, is missing, which poses a problem because agents may change to public transport as soon as a street node meets a station, regardless of the fact that there may be no parking space given. Yet, this specific information is altogether rare, looking at open data resources of the city of Vienna. Surface parking is generally widespread but is newly regulated and time-dependent in availability. Solving this task may be challenging for a multi-modal transportation simulation. For our aims and purposes, this level of detail is not adequate.
- It is inherent to models that reality cannot be depicted sufficiently. Our sub-district commuter model may only output travel times between zones as depending on randomly selected start or end vertices. Exact distribution of commuters within these sub-districts is unknown. Therefore, we cannot produce better results than our background data allows us to, concluding that a finer-level simulation is useless.
- So far, we did not consider time schedules or passage times in public transport; neither did we enable changes within this network itself due to complex model building, big data volumes and processing time. Especially for public transport, complex travel time is relevant [11]. It combines waiting times, changing times, egress times, travel time on board, etc. Egress describes the time needed to access public transport stations, i.e., a commuter has to walk from their residence to a station, wait for a train, get on the train, travel for some time, get off the train and finally, walk from the station to their workplace.
- In this research we meet the general problem if a multi-modal shortest path algorithm may produce "optimal" results. Service quality and quantity in Vienna is high, comprising of many different routes of almost the same trip duration. As said before, representing a highly advanced simulation like that is not our major goal.

IV. CONCLUSION AND FUTURE WORK

Our work-in-progress contribution addresses interesting up-to-date topics in the field of sustainable urban planning as concerned with future needs for de-carbonization. Automobile transportation, widely used in commuter traffic today, obviously has some advantages as compared to public transport. These are: lack of egress, changing and waiting times, little to no body energy requirements, comfort, the option to store luggage and, most of all: constant, unscheduled availability while offering high speed travel. Automobile transport is very time efficient. Even more so, its manifold toxic emissions and its sprawling effects on functional, densely organized urban structures are at opposites with the imperative of restrictive environmental policies, necessary to avoid unpredictable global climate change. Therefore, our research is not aiming at solutions to technical optimization problems but rather poses the general question of what prize cities and their inhabitants would need to pay if seriously considering a major turn towards sustainability.

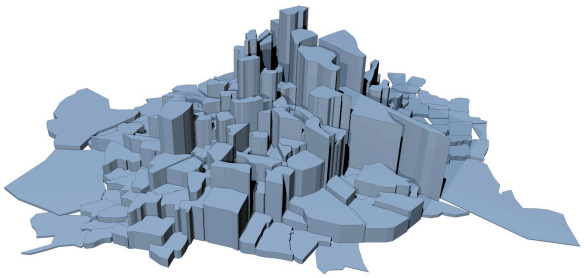


Figure 3. Utilization of the 281 zones of Vienna by automobile commuter through-traffic.

In Figure 3, the zones most frequently traversed by status quos' commuters using automobiles, are depicted by elevation. One of the main highways, passing Vienna from the middle-north to the southeast of its border, is clearly visible. The same utilization, elaborated for public transport users of our baseline model, would look quite different and, would be much more agreeable.

Our contribution offers a smooth application of the shortest-path algorithm as applied to a multi-modal transportation network under the premises of commuters switching to low-carbon transport modes. We can show a nice visual comparison of transport modes routes with regard to travel durations. Figure 4 shows an alternative path per

 ISBN: 978-1-61208-308-7

public transport, found for a current route traveled by individual traffic. Accepting only mode shifts to less polluting transport modes, we highlight the solution space of de-carbonization in commuter traffic, already available today.

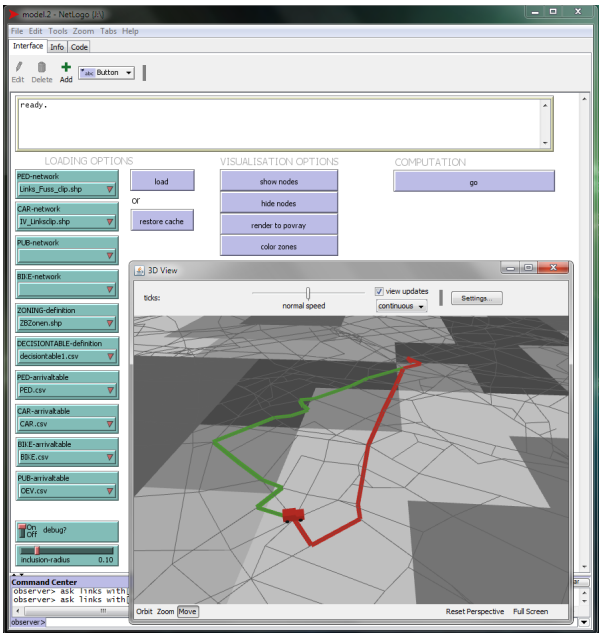


Figure 4. Utilization of the 281 zones of Vienna by automobile commuter through-traffic.

ACKKNOWLEDGEMENTS

We want to thank the Municipal Department 18 of the City of Vienna (Urban Development and Planning) for the provision of the Viennese Transport Model of the City of Vienna.

REFERENCES

[1] J.R. Kenworthy, “Energy use and CO2 production in the urban passenger transport systems of 84 international cities: findings and policy implications”, in: Urban Energy Transition, P. Droege, Ed., Amsterdam: Elsevier, 2008, pp. 211-236.

[2] P.W.G. Newman and J.R. Kenworthy, “The land use-transport connection: An overview”, in: Land Use Policy, vol.13/1, pp. 1-22,1996.

[3] G.Franck and M. Wegener, “Die Dynamik räumlicher Prozesse” in: Raumzeitpolitik, D. Henckel and M. Eberling, Eds, Opladen: Leske & Budrich, 2002, pp. 145-162.

[4] United Nations Framework Convention on Climate Change, http//unfccc.int, 30.09.2013.

[5] M. Wegener, “Overview of land-use transport models”, in: Transport Geography and Spatial Systems, Handbook in Transport, vol. 5, D. A. Henscher and K. Button, Eds. Kidlington: Pergamon/Elsevier Science, 2004, pp. 127-146.

[6] C. J. Campbell, “The Rimini Protocol an oil depletion protocol: Heading off economic chaos and political conflict during the second half of the age of oil” in: Energy Policy, vol. 34/12, pp. 1319-1325, 2006.

[7] Magistrat der Stadt Wien (MA 18) Stadtplanung und Stadtentwicklung, “Verkehrsmodell Wien der Stadt Wien”: PTV VISUM Transportation Model, 2001.

[8] A. Schafer, “The global demand for motorized mobility” in: Transportation Research Part A, vol. 32/6, pp. 455-477, 1998.

[9] M.J. Huguet, D. Kirchler, P. Parent, R. W. Calvo, “Efficient algorithms for the 2-way multi modal shortest path problem” in: Electronic Notes in Discrete Mathematics, vol. 41, pp. 431-437, 2013.

[10] Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

[11] K. Walther, “Nachfrageorientierte Bewertung der Streckenführung im Öffentlichen Personennahverkehr” in: Forschungsberichte des Landes Nordrhein-Westfahlen vol. 2356, Opladen: Westdeutscher Verlag, 1973, pp. 9-142.

 ISBN: 978-1-61208-308-7

Evaluation of the Northern Sardinia Forests Suitability for a Wood Biomass CHP System Installation

Geospatial systems for forest biomass estimation

Pier Francesco Orrù, Emanuela Melis, Laura Fais, Francesca Napoli
Mechanical Engineering, Chemistry and Materials Dept.
Università degli Studi di Cagliari
Cagliari, Italy
pforru@unica.it; emymelis@unica.it; laurafais@hotmail.it; ing.francescanapoli@gmail.com

Cristina Pilo, Michele Puxeddu
Ente Foreste della Sardegna
Servizio Innovazione Tecnologica
Cagliari, Italy
cpilo@enteforestesardegna.it; mpuxeddu@enteforestesardegna.it

Abstract— **In this paper, the results of a preliminary feasibility study for the development of a sustainable supply chain, for the efficient production of heat and power in Sardinia, will be presented. The study area involved the state forest of Monte Olia, for which the biomass availability estimation for energy purposes has been carried out. The biomass estimate has been performed by comparing the results of three methods, using geomatics to environmental and forestry data. In particular, they provide a spatial prediction of the annual biomass supply and simulate the temporal availability for energy use.**

Keywords-geographical information systems; forest biomass; wood-energy supply chain; cogeneration

I. INTRODUCTION

The oil reserves will be able to compensate the estimated global demand for about 50 years [1], at the current consumption rates. Expecting a further world population growth and thus the per capita energy consumption, oil stocks will decrease in an even shorter time.

The chance to cope with the future demand for oil and, more generally, energy, will be based on the ability to best manage the stocks of this fuel, to promote the use of renewable energy sources and low-impact technologies for efficient energy production.

With regard to the national energy situation, in 2011, Italy was the tenth country for natural gas and oil imports, and the fourth for natural gas imports [2].

The current energy situation can be summarised on the basis of the provisional data derived from the National Energy Balance, referred to the year 2012 [3]: compared to an import of 86.278 Mt of oil, 5.397 have been produced and 29.173 have been exported, with a gross energy consumption of 63.590 Mt of oil. So, it represents the most important national energy source, followed by gas (also mostly imported). Therefore, it appears that energy consumption is high and there is a strong dependency on non-renewable energy resources imports.

The international efforts to reduce the fossil fuels consumption and to reduce the non-renewable sources dependence led to the enactment of the Kyoto Protocol, ratified by Italy in 2002, with the subsequent adoption of a National Action Plan for the greenhouse gases (GHG) emissions reduction. In the last GHG Inventory Report [4] for the EU-27, even though between 1990 and 2010 there has been a decrease in GHG emissions equal to 15.4% (excluding Land Use, Land Use Change and Forestry), between 2009 and 2010 there has been an increase of 2.4%.

Specifically, we note that in 2010 CO_2 emissions from fossil fuel combustion increased in the EU-27 by 2.8%. Italy's contribution to the total European emissions (Tg CO_2 eq.) rose from 519 to 501 between 1990 and 2010, representing the fourth nation in the EU-15 and EU-27 for the most amount of emissions. In order to reduce GHG emissions and mitigate the climate change, since 1988 the Intergovernmental Panel on Climate Change (IPCC) gives a clear view of the state of art related to the problem and its potential environmental and socio-economic impacts.

Among the possible ways to reduce GHG emissions concerning heat and power production and supply, the IPCC identified, among others, the improvement in energy conversion, transmission and distribution, including cogeneration and efficiency enhancement in energy user demand in various fields [5].

Notably, the cogeneration is a technology that allows meeting the aforementioned conditions for hazardous emissions reduction and for the improvement of energy efficiency, compared to separate production of heat and power.

Between the renewable energy sources that could be used in such systems, forest biomasses have a considerable importance because, if cropped according to sustainability criteria and with and optimization of cutting, concentration and transport phases (in order to minimize fuel consumption and related emissions), they allow effectively integrating the local energy production. In fact, by 2020, in Europe,

 ISBN: 978-1-61208-308-7

biomass will cover 19% of the renewable produced power and 78% of heating/cooling from renewable sources [6].

Currently, the ever growing energy demand does not match self-containment in energy production at a delocalised level; in this sense, the forest biomass may represent a valuable contribution to the achievement of the targets set by the Kyoto Protocol for 2020, according to forest resources sustainability. Sustainable development strictly connects all the ecological and ecosystemic components with the usability of forests in terms of leisure and use of woody and non-woody products; this is strongly interconnected with the socio-economic and cultural heritage of a territory. Is along these lines that, in 1993, the Ministerial Conference on the Protection of Forests in Europe came to a definition of sustainable forest management: "Sustainable forestry is the management and use of forests and forested areas in a way and at a pace which allows the preservation of their biological diversity, productivity, regeneration ability, vitality, as well as their capability of fulfilling relevant ecological, economic, and social functions at local, national, and global levels now and in the future, in a way which does not damage other eco-systems" [7]. From that viewpoint, where overexploitation of forests, repeated burning, extreme events and infestations impoverished the local forests, biomass removal sustainability leads to a limited and cautious use of the woody resources. The forests productivity is based on complex processes, which must be taken into account in the models for estimating the biomass for energy purposes.

For the exploitation of forest biomass for energy use it is necessary to assess not only the current availability of biomasses, but also its stability over time; in fact, the fuel supply fluctuations can also cause relevant problems on the payback time of a wood-energy chain. This type of supply chains have been developed thanks to the increasing awareness by the scientific world, institutions and industry of the fossil fuels consumption and the environmental emissions associated with their combustion.

In Italy, the commonly used biomass for heat and power production consists mainly of solid biomass derived from forestry and agriculture, agro-industrial residues, biogas and bioliquids [8].

To comply with the sustainability and efficiency criteria, a wood-energy chain has to satisfy certain characteristics: the biomass must be present in sufficient quantities in order to feed a cogeneration plant, the maximum distance from the sampling sites to the system must be less than 70 km (criterion of short chain) and the plant must be able to meet the characteristics imposed by the type of user and of biomasses.

The preliminary feasibility study for the Monte Olia state forest (North-eastern Sardinia) aimed to define the possibility of setting an efficient and sustainable wood-energy supply chain, based on the previously discussed criteria, using the Geographical Information Systems for the spatial estimation of biomass for energy purposes and the prediction of its annual availability.

Firstly, the characteristics of the study area will be presented; then, the three estimation methods applied to the territory will be explained; finally, the results will be discussed.

II. CASE STUDY

A. General Description of the Forest Resources and Territory

The study area concerns the public forest of Monte Olia, located in the North-eastern part of Sardinia (Italy) and it is part of the Forest Complex of Alta Gallura-Buddusò (10887 ha); the forest is managed by Ente Foreste della Sardegna, regional organism, whose mission is to protect, develop and promote Sardinian forests and wildlife. Between the major functions of Ente Foreste della Sardegna, there is also the involvement in research and studies aimed at the development of eco-friendly production activities that are complementary and related to forest management.

The Monte Olia state forest occupies almost 2300 hectares and is characterised, from a geological perspective, by Palaeozoic granite; in fact the area is in the central part of the Corsica-Sardinia batholith, one of the largest European intrusive complexes [9].

The soils of Gallura are generally poorly evolved and shallow. In fact, for example, plowing and repeated use of fires for new pastures creation in *Quercus suber* forests caused a significant reduction of the organic matter in soils. Particularly, in cork production areas the ectorganic horizons are well developed, while in the woody areas primarily used for grazing, the ectorganic horizons are very poorly developed [10].

The vegetation of the state forest has been significantly influenced by repeated fires; the original holm oak mesophilic forest remains in few areas in the valleys, characterised by covers lower than 20-25%, with a dense undergrowth of arbutus, heather, lavender and cistus.

The most consistent group of artificial formations dates back to about 80 years ago, and consists of a high forest of *Pinus pinea*, with dense undergrowth of *Quercus Ilex* and *Quercus Suber*. For the most recent reforestations (1990 and 1991), *Quercus ilex*, *Quercus suber*, *Quercus pubescens* had been planted as well as *Pinus pinea*, *Pinus halepensis*, *Pinus pinaster*, *Pinus nigra ssp*. Laricio. In the eastern part of the state forest pure reforestations of *Quercus Suber* had been done [11].

B. Estimation of the Sustainable Allowable Biomass

For the estimation of forest biomass for energy purposes, reference has been made to three methods:

- Forestry and Environment Regional Plan of Sardinia Region (PFAR) [12],
- Barbati A., Corona P., Mattioli W. and Quatrini A. [13],

 ISBN: 978-1-61208-308-7

- Nocentini S., Puletti N. and Travaglini D. [14],

and the results have been compared, in order to define the most appropriate method with respect to the case study.

Specifically, the PFAR method has been used for a rough estimate of Sardinian availability of forest biomasses; the Barbati A., Corona P., Mattioli W. and Quatrini A. method has been recently proposed in Italy for the Alta Valle dell'Aniene (Lazio), which adapts at a local level the criteria adopted from [15] for reducing the environmental pressures and it considers the current increments of forests.

The method developed by Nocentini S., Puletti N. and Travaglini D. has been developed for high forests of Mediterranean conifers in Tuscany and it takes into account the minimum forest stock.

In order to apply the three selected methods, the following georeferenced Gauss-Boaga (west zone)/Roma 40 shapefiles have been used:

a. Land use map of the Monte Olia state forest: woodland classes are identified by areas > 2000 m^2 with a width >20 m and a coverage > 20%. The minimum mapped unit is 2000 m^2.

b. Forest roads.

c. Spot elevations and contour lines.

Within the preparation of the detailed forestry plans of the state forest, the maps of land use and forest roads have been made in 2012 by assignment to Italian forestry companies. These data have been provided by Ente Foreste della Sardegna, while the two layers containing the elevation data (elevation points, contour lines) are available for free online [16].

Only the polygons of forest have been considered, for each woody class the areas have been derived (hectares).

The distribution of the land use classes is shown in Table I.

TABLE I. LAND USE SURFACES (PERCENTAGE OF THE MONTE OLIA STATE FOREST)

Land Use Class	Surface (% of the state forest)
waters	0.21
shrubs	23.91
conifers	39.14
deciduous broadleaves	0.34
rupestrian woodland	1.72
evergreen broadleaves	7.95
firebreaks	1.40
crops	0.06
maquis	16.74
pasture lands	1.64
failed reforestation	0.75
rocks	6.11
urban fabric	0.03

The woodlands occupy less than a half (47%) of the Monte Olia state forest and consist almost entirely of conifers (82%), while the remaining part is covered by evergreen broadleaf woods (only the 0.3% of the total area is covered by deciduous broadleaves).

PFAR method - The estimation of the available biomass is different for broadleaves in the state forests and conifers in the state forests.

- Broadleaves in the state forests

  The average increment of 2.14 $m^3 ha^{-1} yr^{-1}$ has been applied to the woody polygons; the surfaces have been multiplied with the usage coefficient of 20%.

In order to compare the findings of the three methods and to obtain the potential allowable cut of biomass (tons of dry matter per year), it has been necessary to multiply the result by the Wood Basic Density (WBD) and Biomass Expansion Factor (BEF) coefficients [17], whose values are reported in Table II. The BEF factor expands the growing stock volume to the volume of aboveground woody biomass (aboveground biomass/growing stock); the WBD coefficient allow converting the fresh volume of timber wood to dry weight (dry matter, 20% of humidity).

The final values for broadleaves are 79.6 t d. m. yr^{-1}.

- Conifers in the state forests

  A range minimum-maximum for the biomass stock per hectare has been assigned: 170÷200 m^3 ha^{-1}. For each limit of this range the steps are indicated below:

  - The surfaces have been multiplied by the usage coefficient 0.45 and then by the lower or upper limit of biomass per hectare (170 and 200 m^3 ha^{-1} respectively).
  - All the biomass will be cut during the next 20 years: the values obtained at the last point have been divided by 20 years.

  At the end of those steps, we applied the WBD and BEF coefficients and we obtained 2089.7 t d. m. yr^{-1} for the lower limit and 2458.5 t d. m. yr^{-1} for the upper limit.

The total available quantity of biomass for energy uses is between 2169.3 t d. m. yr^{-1} (lower limit condition) and 2538.1 t d. m. yr^{-1} (upper limit condition).

Barbati A., Corona P., Mattioli W. and Quatrini A. method - For the wooded classes indicated in the land use map, the corresponding current increments derived from the National Inventory of Forests and Carbon Sinks (INFC) [18] have been applied to the relative polygons as well as the WBD and BEF coefficients (Table II).

TABLE II. CURRENT INCREMENTS, BIOMASS EXPANSION FACTOR AND WOOD BASIC DENSITY

Forest Classes	*Current Increment ($m^3 ha^{-1} yr^{-1}$)*	*BEF*	*WBD (tons of dry matter per m^3 of fresh volume)*
Conifer	3.4	1.37	0.43
Deciduous Broadleaf	1.3	1.47	0.53
Evergreen Broadleaf	1.3	1.42	0.67

 ISBN: 978-1-61208-308-7

The biomass obtained by the product of current increments, BEF, WBD and surfaces for each of the forest classes is equal to 2096.39 t d. m. yr^{-1}.

Afterwards the removal reduction coefficients have been applied, according to the accessibility of forestry vehicles, in order to estimate the net potential allowable biomass cut.

The forest accessibility is a limiting factor for cutting and skidding tracks and essentially depends on slope and distance from roads; the reduction factors proposed by the model (Table III) have been used, by changing the coefficient value from 0 to 0.25 for the slope class 21% - 30% and distance from roads 500 - 2500 m, since slopes between 20 and 35% and distances from the roads <2500 m are on average accessible [19]; for distances from roads >2.5 km, cutting and skidding costs are prohibitive and the coefficient is 0 for all slopes.

We subsequently proceeded to the generation of DEM (from the elevation values a.s.l. of the two layers of quoted points and contour lines), whose spatial resolution is 20 m × 20 m. The slopes map has been extracted from those raster files.

TABLE III. REDUCTION COEFFICIENTS

Slope class (%)	*Distance from forest roads (m)*			
	0-150	***150-500***	***500-2500***	***>2500***
0-20	0.75	0.75	0.75	0
20-30	0.5	0.5	0.25	0
30-50	0.25	0	0	0
>50	0	0	0	0

With regard to slopes, the Monte Olia forest has slopes greater than 30% for the most part (46.7%); about one third of the territory is in the slope class 0% - 20% (30.7%) and the rest falls within the class 20% - 30% (22.6%).

The distance from roads map (Figure 1) has been created by applying the *Cost Distance* algorithm (implemented in the ArcGIS software, which has been used for the biomass estimation), considering the forest roads as input.

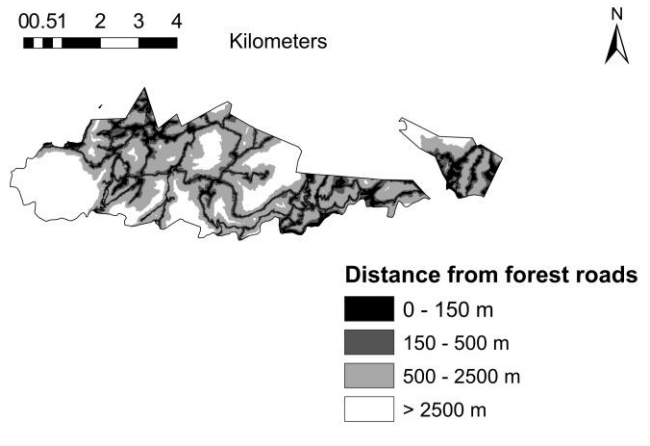


Figure 1. Map of the distances from roads, Monte Olia

The algorithm allows obtaining a map of the cumulative distances from forest roads with respect to a cost surface, the slopes map. We have decided to not apply the *Euclidean Distance* algorithm, since the territory has highly accentuated slope variations, which should be taken into account in terms of feasibility of cutting and skidding.

The raster files of slopes and distance from roads have been combined, in order to assign the coefficients of Table II to the woody cells.

The map of potential allowable biomass cut before the accessibility criterion has been multiplied by the reduction coefficients map, using the *Map Algebra*; the algorithm multiplies cell by cell the value of biomass with the reduction coefficient. The final output is a map of the net allowable biomass cut, considering the limitations due to accessibility (t d. m. yr^{-1}).

For the Monte Olia state forest 719.5 t d. m. yr^{-1} have been obtained.

The final biomass estimation map shows a limited availability of forest biomasses, with meager and fragmentarily distributed quantities, with respect to the result obtained by the application of the PFAR method.

Nocentini S., Puletti N. and Travaglini D. method - This model takes into account the theory of the systemic silviculture [20][21][22] and the notion of the Safe Minimum Standard [23].

The method is based on the concept according to which it is possible to cut the biomass if the real stock P_r is greater than the minimum stock P_m of 20% (P_r/P_m=1.2), with a removal rate depending on this ratio.

First of the application of the method, the areas with slopes >35% and distant more than 2.5 km from forest roads have been omitted.

Subsequently, the real stocks have been assigned to the remaining woodland polygons: reference has been made to [18]. Such data must be referred to an initial time t_0 for the estimate: it has been set at the year 2007, which coincides with the end of the phase 3+ of the last INFC [18].

Starting from t_0, the iterations of the method have been performed:

-at the time t_0 the Pr is equal to the stock of INFC;

-at the time t_1, $Pr_1=Pr_0$+the growing rate (m^3 ha^{-1} yr^{-1}).

-In the most general condition, we verify Pr_{x-1}/P_m >1.2: if true, the removal rates indicated in Table IV can be applied; if it is false, we continue by doing the comparison at the time t_x. If at the time t_{x-1} a certain amount of biomass has been removed, at the time t_x it has to be subtracted to the comparison.

For the case study, the only forest class which has a value of Pr_0 similar to P_m (equal to 100 for eliophilous species [14]) is that of the conifers, so the iterations have been done solely for this class (Figure 2), starting in 2007 and carrying out the calculations for 16 years.

 ISBN: 978-1-61208-308-7

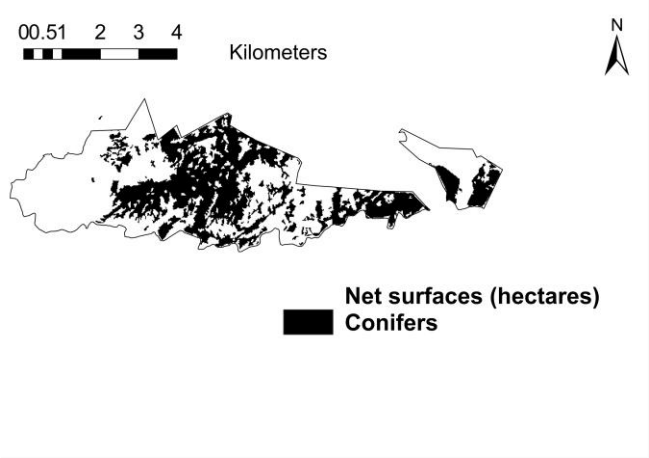


Figure 2. Map of net surfaces of conifers, Monte Olia

The results are reported in Table V and Table VI.

TABLE IV. REDUCTION COEFFICIENTS

Pr/Pm	*Annual Removal (%)*
>2	1.5
1.8÷2	1.25
1.6÷1.8	1
1.4÷1.6	0.75
1.2÷1.4	0.5

TABLE V. ITERATIONS FROM THE FIRST TO THE EIGHTH YEAR

T	Year	Pr	Pr/Pm	Value	Removal
t_0	2007	95.9	0.96	<1.2	NO
t_1	2008	99.3	0.99	<1.2	NO
t_2	2009	102.7	1.03	<1.2	NO
t_3	2010	106.1	1.06	<1.2	NO
t_4	2011	109.5	1.10	<1.2	NO
t_5	2012	112.9	1.13	<1.2	NO
t_6	2013	116.3	1.16	<1.2	NO
t_7	2014	119.7	1.197	<1.2	NO

TABLE VI. ITERATIONS FROM THE NINTH TO THE SIXTEENTH YEAR

Year	Pr	Pr/Pm	Comparison	Removal Rate ($m^3 ha^{-1} yr^{-1}$)	Removal (Fresh volume) $m^3 yr^{-1}$	t d. m. yr^{-1}
2015	123.1	1.23	1.2 ÷ 1.4	0.005	442.1	260.8
2016	125.9	1.26	1.2 ÷ 1.4	0.005	452.3	266.7
2017	128.7	1.29	1.2 ÷ 1.4	0.005	462	272.6
2018	131.4	1.31	1.2 ÷ 1.4	0.005	471.9	278.4
2019	134.2	1.34	1.2 ÷ 1.4	0.005	481.7	284.2
2020	136.9	1.37	1.2 ÷ 1.4	0.005	491.5	290
2021	139.6	1.40	1.2 ÷ 1.4	0.005	501.3	295.7
2022	142.3	1.42	1.4 ÷ 1.6	0.0075	766.5	452.2

Table V shows that, for the first eight years, it is not possible to cut biomass in the Monte Olia state forest, due to the fact that the real biomass stock is less than the minimum stock.

Between 2015 and 2022 the biomass cutting will be possible and the average availability of biomass for that period is around 300 t d. m. yr^{-1}; the maximum value is 452 t d. m. yr^{-1} of total allowable timber.

From the application of this method, it appears a very limited biomass availability for energy purposes; this allows us to assert that these quantities may be used to feed a small cogeneration plant.

By comparing the results obtained by the implementation of the three above discussed methods for the Monte Olia state forest, we noticed that:

- The first method (PFAR) [12] provided a very huge quantity of biomass. This model is not good for a real estimation of the biomass availability; in fact it doesn't consider any constraints about biomass removal.

- The Barbati A., Corona P., Mattioli W. and Quatrini A. method and the Nocentini S., Puletti N. and Travaglini D. method lead to lower values; in order to decide which has to be taken into account, in order to design a wood-energy supply chain for a cogeneration system installation close to the study area, it is important to analyse them from the point of view of the forest stands and regional forest management.

Specifically for the state forest of Monte Olia, the guidelines of the silvicultural interventions are based on the systemic silviculture and the minimum forest stock [14][22]: so, the available biomass for energy purposes is strictly related to those principles. The Barbati A., Corona P., Mattioli W. and Quatrini A. method considers the current increments instead of the forest stocks: it may occur that, by applying this model, the real forest stock is lower than the minimum stock, but if we cannot take it into account, the estimation of the available biomass does not correspond to the real condition.

Furthermore, the two methods differ for the considered forest classes: in the Barbati A., Corona P., Mattioli W. and Quatrini A. model we use all the woody classes (conifers as well as broadleaves); in the Nocentini S., Puletti N. and Travaglini D. method, we must take into consideration only the classes which comply with the condition $P_r/P_m > 1.2$.

From the comparison between the three methods, it is clear that the method proposed by Nocentini S., Puletti N. and Travaglini D. is the most appropriate, because of the

 ISBN: 978-1-61208-308-7

abovementioned reasons and also because it provides the lowest and most precautionary value.

III. CONCLUSION

The preliminary feasibility study of a sustainable supply chain for the efficient energy production in a cogeneration plant, using the Monte Olia forest biomasses, has allowed us to determine if the biomass quantities are sufficient for their use in cogeneration plants and which is the most appropriate estimation model for the case study.

The quantification of the forest biomass for energy purposes in the study area has shown a very limited biomass availability, which enables to install only a small size cogeneration plant close to the area.

By the application of the three chosen models and the comparison between their results, it has been possible to verify the significant differences and to select the most suitable methodology for the estimation of forest biomass for energy uses. The Nocentini S., Puletti N. and Travaglini D. method could be used not only for Monte Olia, but also for other public forests which have similar conditions.

The above mentioned method has been effectively used to obtain a simulated situation of the annual biomass removal for the next nine years.

The research will continue by developing the supply chain: laboratory analyses will be conducted on the most relevant forest species, in order to know the fuel characteristics; an energy audit is being carried on a service building within the study area and an economic evaluation of the supply chain will be performed.

ACKNOWLEDGMENT

The research has been funded by Ente Foreste della Sardegna through a three-year agreement.

The research has been carried out within the "Master And Back" Program of the Autonomous Region of Sardinia.

Gratefully acknowledges Sardinia Regional Government for the financial support (P.O.R. Sardegna F.S.E. Operational Programme of the Autonomous Region of Sardinia, European Social Fund 2007-2013 - Axis IV Human Resources, Objective l.3, Line of Activity l.3.1 "Avviso di chiamata per il finanziamento di Assegni di Ricerca").

REFERENCES

[1] R. C. Duncan and W. Youngquist, Encircling the peak of world oil production. Natural Resources Research, vol. 8, n. 3, 1999, pp. 219-232.

[2] ENI Company, [online] Available at <http://www.eni.com/world-oil-gas-review-2012/static/pdf/wogr-2012.pdf> [Accessed 19 August 2013].

[3] Ministry of Economic Development of the Italian Republic, [online] Available at <http://dgerm.sviluppoeconomico.gov.it/dgerm/ben.asp> (file referred to 2012, provisional data uploaded 26 April 2013) [Accessed 19 August 2013].

[4] EEA, Annual European Union greenhouse gas inventory 1990–2010 and inventory report 2012 - Technical report no.3, 2012.

[5] Intergovernmental Panel on Climate Change, IPCC Special Report on Renewable Energy Sources and Climate Change Mitigation. Prepared by Working Group III of the Intergovernmental Panel on Climate Change. Cambridge, Cambridge University Press, United Kingdom and New York, NY, USA, 2011, p. 174.

[6] C. Panoutsou and K. Maniatis, Biomass futures: Estimating the role of sustainable biomass for meeting the 2020 targets and beyond. 2013, Biofuels, Bioproducts and Biorefining 7 (2), pp. 97-98.

[7] Helsinki Resolution H1, Second Ministerial Conference on the Protection of Forests in Europe, 1993.

[8] N. Scarlat, J. F. Dallemand, V. Motola, and F. Monforti-Ferrario, Bioenergy production and use in Italy: recent developments, perspectives and potential. Renewable Energy 57, 2013, pp. 448-461.

[9] G. Oggiano and A. Di Pisa, Introduction to the Sardinia geologic evolution. Reports of the Seminary Sciences Faculty of the University of Cagliari, Appendix Vol. 71, Issue 2, 2001.

[10] A. Vacca, Effect of land use on forest floor and soil of a Quercus suber L. forest in Gallura (Sardinia, Italy). Land Degrad. Develop., 11 2000, pp. 167-180.

[11] Ente Foreste della Sardegna, [online] Available at <http://www.sardegnaambiente.it/j/v/152?s=8661&v=2&c=1653&t=1> [Aceessed 19 August 2013].

[12] Autonomous Region of Sardinia, Department of Enviromental Protection, Environment and Forestry Regional Plan, Appendix 3, 2007 [online] Available at <http://www.regione.sardegna.it/documenti/1_73_20080129175721.pdf> [Accessed 19 August 2013].

[13] A. Barbati, P. Corona, W. Mattioli, and A. Quatrini, Forest biomass for heat energy generation: an analysis model for the Upper valley of Aniene River. Italian Journal of Forest and Mountain Environments, 67 (4), 2012, pp. 329-336.

[14] S. Nocentini, N. Puletti, and D. Travaglini, Planning and sustainable use of forest biomass in the forest-energy chain: a methodological proposal. Italian Journal of Forest and Mountain Environments, 66 (4), 2011, pp. 293-303.

[15] EEA, How much bioenergy can Europe produce without harming the environment? EEA Report no. 7, 2006.

[16] Autonomous Region of Sardinia, [online] Available at <http://www.sardegnageoportale.it/catalogodati/download/> [Accessed 19 August 2013].

[17] ISPRA, Italian Greenhouse Gas Inventory 1990-2007. National Inventory report, 2009.

[18] P. Gasparini and G. Tabacchi (edited by), The National Inventory of Forests and forest Carbon stocks INFC 2005. Second Italian national forest inventory. Methods and Results. Ministry of Agricultural, Food and Forestry Policies; State Forestry Corps. Council for Research and Experimentation in Agriculture, Research Unit for monitoring and forest planning. Edagricole-Il Sole 24 ore, Bologna, 2011.

[19] O. Ciancio, P. Corona, M. Marinelli, and D. Pettenella (edited by), Evaluation of Forest Fire Damages in Italy. Firenze: Tipografia Coppini, Oct. 2007, p. 39.

[20] O. Ciancio and S. Nocentini, Systemic silviculture: scientific and technical consequences. Italian Journal of Forest and Mountain Environments, 51, 1996, pp. 112-130.

[21] O. Ciancio, P. Corona, M. Marchetti, and S. Nocentini, Systemic forest management and operational perspectives for implementing forest conservation in Italy under a pan-European framework. Proceedings of the XII World Forestry Congress, Vol.B, Outstanding paper, Level 1, Québec City, 2003, pp. 377-384.

[22] O. Ciancio and S. Nocentini, Biodiversity conservation and systemic silviculture: Concepts and applications, Plant Biosystems – An International Journal Dealing with all Aspects of Plant Biology: Official Journal of the Società Botanica Italiana, 145:2, 2011, pp. 411-418.

[23] M. A. Toman, The difficulty in defining sustainability. Resources (published by Resources for the future), no. 106 (winter), pp. 3-6.

 ISBN: 978-1-61208-308-7

Developing a Simulation Model for a Level of Usage

Andrew Greasley
Aston Business School
Aston University
Birmingham, UK
e-mail: a.greasley@aston.ac.uk

***Abstract*— This paper provides a framework categorising the level of usage of a simulation model and relating this to three key activities in the simulation process of model development, model interaction, and model integration. The aim of the framework is to clarify how the level of usage will decide both the nature of the information derived from the simulation study and the development activities that are required of the model builder. Further work is required in order to determine if in practice the level of usage is determined by the level of information required or other factors, such as a lack of user skills is preventing a higher level of usage of the technique in the organisation.**

Keywords-simulation; methodology; development

I. INTRODUCTION

There are many texts available on the topic of discrete-event simulation modeling. Some texts focus on technical and statistical aspects [1,2], other texts focus on the application of simulation for process analysis [3,4], other texts provide tutorials on particular simulation software platforms, such as ARENA [5,6,7], and other texts take the reader through the steps involved in undertaking a simulation study [8,9,10,11,12,13]. However, all of these different approaches generally provide little guidance on the form of the simulation that is appropriate to the needs of the study. This should be considered in terms of viewing simulation as a tool which provides information to assist in decision making. It follows that in order to assist the decision-making process it is not always necessary to undertake all the stages of a simulation study. For instance, the development of the process map may be used to help understanding of a problem and consequently, no further model development is necessary.

The paper will provide an overview of proposed forms of simulation development. Four forms are identified and labeled on a continuum of levels of usage of the simulation technique. The forms are defined by three variables related to key aspects of the simulation project effort of development, interaction, and integration. There is then a discussion of these forms and an indication of further research required in order to validate these forms.

II. DETERMINING THE LEVEL OF USAGE OF THE SIMULATION MODEL

An important aspect in the process of building a simulation model is to recognize that there are many possible ways of modelling a system. Choices have to be made regarding the level of detail to use in modelling processes and even whether a particular process should be modelled at all. The way to make these choices is to recognize that before the model is built, the objectives of the study must be defined clearly. It may even be preferable to build different versions of the model to answer different questions about the system, rather than build a single 'flexible' model that attempts to provide multiple perspectives on a problem [14]. This is because two relatively simple models will be easier to validate and thus, there will be a higher level of confidence in their results than a single complex model.

The objective of the simulation technique is to aid decision making by providing a forum for problem definition and providing information on which decisions can be made. Thus, a simulation project does not necessarily require a completed simulation model to be a success. At an early stage in the project proposal process the analyst and other interested parties must decide the role of the model building process within the decision-making process itself. Thus, in certain circumstances as stated earlier the building of a simulation model may not be necessary. However, for many complex interacting systems (i.e., most business systems), the model will be able to provide useful information (not only in the form of performance measures, but indications of cause and effect linkages between variables), which will aid the decision making process. Table 1 provides a framework which links four categories of usage of the simulation model, namely 'problem definition', 'demonstration', 'scenarios' and 'on-going decision support' with three key aspects of the simulation process. These three key aspects are the level of development of the model, the level of interaction between the model and user, and the level of integration of the model with its data set that would be implied by the level of usage.

 ISBN: 978-1-61208-308-7

TABLE I. LEVELS OF USAGE OF A SIMULATION MODEL

	Level of Usage			
	Problem Definition	**Demonstration**	**Scenarios**	**On-going Decision Support**
Level of Development	Process Map	Animation	Experimentation	Decision Support System
Level of Interaction	None	None Simple Menu	Menu	Extended Menu
Level of Integration	None	Stand-alone	Stand-alone Database	Stand-alone Database Real-Time Data

The levels of usage categories are defined as follows:

A. Problem Definition

One of the reasons for using the simulation method is that its approach provides a detailed and systematic way of analysing a problem in order to provide information on which a decision can be made. It is often the case that ambiguities and inconsistencies are apparent in the understanding of a problem during the project proposal formulation stage. It may be that the process of defining the problem may provide the decision makers with sufficient information on which a decision can be made. In this case, model building and quantitative analysis of output from the simulation model are not required. In terms of development, the outcome from this approach will be a process map of the system. As no model is constructed the level of model interaction and model integration categories are not relevant.

B. Demonstration

Although the decision makers may have an understanding of system behaviour, it may be that they wish to demonstrate that behaviour to other interested parties. This could be to internal personnel for training purposes or to external personnel to demonstrate capability to perform to an agreed specification. The development of an animated model provides a powerful tool in communicating the behaviour of a complex system over time. Here, the model should be developed to such as level as to enable the presentation of a realistic animation of the process. This will reveal the mechanics of process behaviour over time but will not provide a numerical indication of performances which are provided by the experimentation analysis undertaken in higher levels of usage. In terms of interaction, a simple menu system may be useful in providing a convenient method of altering parameters for the animation.

However, there is unlikely to be a need for integration with external data sets, due to the lack of scenario analysis when used in the demonstration mode.

C. Scenarios

This category of usage can be related to the ‘classic’ or ‘textbook’ use of the simulation method. Here, the model is developed, validated and scenario analysis conducted. Results are presented of performance measures of interest usually in the form of confidence intervals. In the scenario category, the model is used to solve a number of pre-defined problems but is not intended for future use. For this reason, in terms of the level of interaction, a menu system allowing change of key variables may be appropriate. In terms of integration, the simulation may use internal data files or ideally be linked to external databases.

D. On-going Decision Support

The most fully developed simulation model must be capable of providing decision support for a number of problems over time. This requires that the model be adapted to provide assistance to new scenarios as they arise. The menu system will need to provide the ability to change a wider range of variables for on-going use. The level of data integration may require links to company databases to ensure the model is using the latest version of data over time. Links may also be required to real-time data systems to provide on-going information on process performance. If it is envisaged that the client will perform modifications to the simulation model after delivery, then, the issue of model re-use should be addressed. Re-use issues include ensuring detailed model code documentation is supplied and detailed operating procedures are provided. Training may also be required in model development and statistical methods.

 ISBN: 978-1-61208-308-7

III. DISCUSSION

A simulation modelling project can use extensive resources, both in terms of time and money. Although the use of simulation in the analysis of a one-off decision, such as a major investment appraisal, can make these costs low in terms of making the correct decision, there is a need to ensure the correct level of usage is chosen in alignment with the information required for the decision. Indeed, it has been noted that it is not a requirement of a simulation modelling exercise that a model is actually built, but qualitative outcomes from the process mapping stage, for example, could generate useful knowledge. This elicitation of knowledge through the process of conducting a simulation study rather than simply an observation of model results is termed 'simulation for facilitation' by Robinson [15]. On the other hand, the advantages of a higher level of usage may be considerable in terms of a greater amount of information gained as the level of usage is increased and there is also evidence that developing a model with on-going decision-support capabilities increases model confidence and acceptance particularly among non-simulation experts [16]. However, the consequences of developing a model with a high level of usage in terms of model development, interaction and integration should be considered. The use of simulation for on-going support is particularly challenging and it is thus important that during the project proposal stage that elements are incorporated into the model and into the implementation plan that assist in enabling the model to provide on-going decision support. Aspects include ensuring that simulation users are aware at the project proposal stage that the simulation is to be used for on-going decision support and will not be put to one side once the immediate objectives are met. Also, ensuring technical skills are transferred from simulation analysts to simulation users will enable understanding of how the simulation arrives at results and its potential for further use in related applications.

To assist in the identification of the form of simulation that users adopt, a framework containing four levels of usage of simulation has been presented. It is proposed that a survey be conducted to establish the validity of these forms and the proportion of use of each of the four levels of usage. The aim will be to establish if there is a match between the level of usage and the information needs of the decision being taken or if there are other factors, such as lack of skills, impacting on the choice of level of usage.

IV. CONCLUSION

This paper aims to highlight an important aspect of conducting a simulation study, namely, the level of usage of the simulation model. This issue is important because it will decide both the nature of the information derived from the study and the development activities that are required by the model builder. Further work is required in order to validate the level of usage by practitioners and their implications for the simulation process. An important question is to determine why the different forms are utilised. For instance, is the level of usage determined by the level of information required or is it that the lack of user skills is preventing a higher level of usage of the technique in the organisation?

REFERENCES

[1] A.M. Law and W.D. Kelton, Simulation Modeling and Analysis, Third Edition, Singapore: McGraw-Hill, 2000.

[2] L.M. Leemis and Stephen K. Park, Discrete-Event Simulation: A First Course, Pearson Education, 2006.

[3] H.J. Harrington and K. Tumay, Simulation Modelling Methods: To reduce risks and increase performance, McGraw-Hill, 2000.

[4] M. Laguna and J. Markland, Business Process Modeling, Simulation and Design, Pearson Education, 2005.

[5] M.D. Rossetti, Simulation Modeling and Arena, Wiley, 2010.

[6] T. Altiok and B. Melamed, Simulation Modeling and Analysis with Arena, Academic Press, 2007.

[7] W.D. Kelton, R.P. Sadowski, and D.T. Sturrock, Simulation with Arena, Fourth Edition, McGraw-Hill, 2007.

[8] S.Robinson, Successful Simulation: A Practical Approach to Simulation Projects, McGraw-Hill, 1994.

[9] R. McHaney, Computer Simulation: A Practical Perspective, Academic Press 1991.

[10] M. Pidd, Computer Simulation in Management Science, Fifth Edition, Wiley, 2004.

[11] S. Robinson, Simulation: The Practice of Model Development and Use, Wiley, 2004.

[12] A.F. Seila, V. Ceric, and P. Tadikamalla, Applied Simulation Modelling, Thomson, 2003.

[13] A. Greasley, Simulation Modelling for Business, Hants: Ashgate Publishing Ltd., 2004.

[14] M. Pidd, Tools for Thinking: Modelling in Management Science, Third Edition, Wiley, 2009.

[15] S. Robinson, "Modes of simulation practice: approaches to business and military simulation", Simulation Modelling Practice and Theory, no. 10, 2002, pp. 513-523.

[16] D.J. Muller, "Simulation: What to do with the Model afterward", Proceedings of the 1996 Winter Simulation Conference, Society for Computer Simulation, 1996, pp. 729-733.

 ISBN: 978-1-61208-308-7

A CC2420 Transceiver Simulation Module for ns-3 and its Integration into the FERAL Simulator Framework

Anuschka Igel and Reinhard Gotzhein
Networked Systems Group
University of Kaiserslautern, Germany
{igel,gotzhein}@cs.uni-kl.de

***Abstract*—Simulation is a common approach to assess the functional and non-functional behavior of protocols in wireless sensor networks. In these networks, the CC2420 transceiver is a frequently used communication platform. To make this platform available for simulation purposes, we have developed a CC2420 simulation module for ns-3, a well-known discrete-event network simulator targeted primarily for research and educational use, using an existing CC2420 module for its predecessor ns-2 as starting point. In this paper, we report on this development and, in particular, several functional enhancements of the existing CC2420 module. Furthermore, we present the integration of ns-3 and the CC2420 module into FERAL, a generic simulator framework for the rapid coupling of diverse simulators. Finally, we present results of simulation experiments where we have used the CC2420 module, both stand-alone ns-3 simulations and simulations where ns-3 is a simulator component of the FERAL framework. These experiments show that the CC2420 simulation module is fully operational in the ns-3 context, and that the integration into FERAL provides additional degrees of freedom especially in the early development stages, where abstract models, e. g., Simulink or SDL models, are used to specify system behavior.**

Keywords-CC2420 simulation module; ns-3; net device; simulator framework

I. INTRODUCTION

Nowadays, Mobile Ad-hoc NETworks (MANETs) consisting of wireless sensor nodes become more and more important. Common application areas are the collection of environmental data in inaccessible areas or health monitoring. These networks are characterized by a lack of fixed infrastructure. Due to node mobility, they are usually restricted concerning power supply. Therefore, the use of energy-efficient hardware is crucial. A common transceiver module used for such nodes is TEXAS-INSTRUMENTS' CC2420 transceiver [1], which is compliant with the IEEE 802.15.4 standard [2]. This standard defines wireless transmissions in low-cost networks, with devices that have a low data rate and low power.

Protocols running on nodes in MANETs can be quite complex and therefore should be evaluated before deployment. Since testbeds are expensive and time-consuming to build, simulations are often used to verify functional as well as non-functional behavior of these protocols. Therefore, simulation capabilities for MANETs whose nodes communicate via CC2420 transceivers are desirable. Network simulators suitable for MANETs already exist, including the well-known network simulator 2 (ns-2) [3] and its successor, the network simulator 3 (ns-3) [4]. A CC2420 module for ns-2 has been developed in [5] and integrated into the simulator C-PartsSim (see also [6]); however, ns-2 is no longer actively developed and has several drawbacks compared to ns-3 (see [7]). For example, ns-2 uses a combination of C++ and the Tool Command Language (TCL), while ns-3 is written entirely in C++. Besides, the existing CC2420 module does not realize important features (e. g., changing certain settings of the transceiver). Therefore, we have developed a CC2420 simulation module with several functional enhancements for ns-3, taking the existing module for ns-2 as a starting point. The decision for ns-3 was especially taken because of its modularity and clean design, which allows using many of the existing simulation components – including applications, protocol implementations, and mobility, loss and delay models – together with the CC2420 simulation module.

In addition to this, we have integrated ns-3 and the CC2420 module (and other ns-3 components) into the Framework for the Efficient simulator coupling on Requirements and Architecture Level (FERAL) [8], thus drawing benefit from using it in combination with other simulators such as Simulink. In this paper, we will use a simulation component for the internationally standardized Specification and Description Language (SDL) [9] to simulate the behavior of nodes, while ns-3 and the CC2420 module are used to simulate the behavior of the medium.

The remainder of this paper is structured as follows: In Section II, we survey related work. In Section III, we describe the development and enhancement of the CC2420 simulation module. Section IV reports on the integration of ns-3 and the CC2420 module into the simulator framework FERAL. Section V presents results of simulation experiments using the CC2420 module. In Section VI, we draw conclusions and elaborate on future work.

II. RELATED WORK

Related work can be divided into two categories, namely simulation approaches for the CC2420 transceiver and the development of simulation modules for ns-3. In this paper, we combine these two aspects and report on the development of a CC2420 simulation module for ns-3.

Several simulation approaches for the CC2420 transceiver are described in the literature. The authors of [10] extended the TinyOS SIMulator (TOSSIM) by an improved wireless propagation model and a radio frequency physical stack based on the CC2420 transceiver. TinyOS is a popular operating

 ISBN: 978-1-61208-308-7

system for wireless sensor networks. Its source code can be directly utilized for TOSSIM, which also considers operating system overhead. Amongst other features, the proposed CC2420 model uses Clear Channel Assessment (CCA), measures the received signal strength and allows configuration of transmission power and channel. The results obtained with this simulator are independent of a concrete processor, but are bound to the TinyOS operating system. Therefore, it cannot be used to evaluate protocols independently of a concrete operating system.

In [11], AvroraZ is presented, which extends the Avrora simulator by a module for simulating the CC2420 transceiver. Avrora is a cycle-accurate instruction-level simulator for AVR microcontrollers. The simulation module includes address recognition, frame acknowledgement, CCA, and several other features. A similar work is presented in [12], where an instruction-level sensor network simulator using a detailed CC2420 simulation model is introduced. This simulator provides a cycle-accurate processor emulation of the ATmega128, which is independent of the operating system, and also models the internal structure of the CC2420 transceiver, e. g., registers and main memory. However, the paper does not provide much detail about the CC2420 implementation. Both of these approaches provide instruction-level simulators, which are not suitable for large networks, because simulations are very time-consuming. Furthermore, these simulators are tied to special processors and are therefore not usable for the generic evaluation of protocols on higher layers.

Numerous simulation modules for ns-3 have been developed by third parties. Among these are routing protocols, e. g., the Ad-hoc On-Demand Distance Vector (AODV) protocol, which was implemented for ns-3 in [13]. A module for the Destination-Sequenced Distance Vector (DSDV) routing protocol was introduced in [14], while [15] presents an IPv6 stack for ns-3. Besides this, an ns-3 framework usable for spectrum-aware simulations was described in [16]. The authors implemented spectrum-aware channel and physical layer models, which provides the possibility to analyze how the performance of protocols on higher layers is affected by the frequency-related aspects of communication on the physical layer.

III. CC2420 Transceiver Simulation Module

The CC2420 transceiver developed by TexasInstruments is a 2.4 GHz IEEE 802.15.4 compliant transceiver with a data rate of up to 250 kbps and 16 channels [1]. It is low-cost, configurable, energy-efficient and works in the unlicensed ISM band. Therefore, it can be used to build up sensor networks.

A. Functionality

The CC2420 simulation module adopts the state machine described in the data sheet [1]. We abstract from aspects like resetting and switching the transceiver on and off as well as the acknowledgement mechanism and overflow respectively underflow detection. Figure 1 shows an excerpt of the CC2420 state machine used for the simulation module.

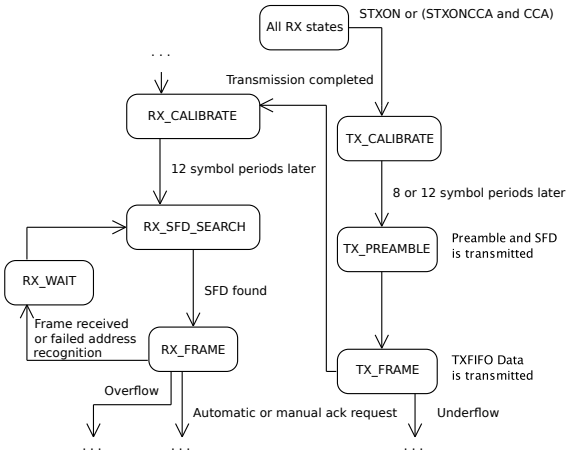


Figure 1. Simulated part of the CC2420 transceiver [1].

The transceiver starts in state RX_CALIBRATE. This phase has a duration of 12 symbol periods, after which the state RX_SFD_SEARCH is entered. Since each symbol encodes 4 bits and the data rate of the transceiver is 250 kbps, one symbol period has a duration of 16 μs. The transceiver uses a synchronization header for symbol synchronization of a received frame. This synchronization header consists of a preamble sequence and a so-called Start of Frame Delimiter (SFD) (the frame format is depicted in Figure 4). The default preamble sequence consists of 4 bytes with value 0x00, while SFD is 0xA7, both compliant with the IEEE 802.15.4 standard [2]. The so-called sync word consists of the last preamble byte (which should be zero for compliance with IEEE 802.15.4) and the SFD byte. Preamble length (i. e., the number of leading zero bytes) and sync word can be configured via special registers, but changing the default values makes the transceiver non-compliant with IEEE 802.15.4. When receiving a frame, the transceiver synchronizes to the zero-symbols and searches for the SFD sequence (the preamble length does not matter here; it is only relevant for sending). The reception of an SFD causes the transceiver to switch to the state RX_FRAME. As soon as the frame is received completely, the state RX_WAIT is taken. When the transceiver is ready to receive another frame, it goes again to the state RX_SFD_SEARCH.

Transmission requests are allowed in all RX states. It is possible to transmit with clear channel assessment (signal STXONCCA) or without (signal STXON). [2] defines three different CCA modes, which are all supported by the CC2420 transceiver. Mode 1 means that the channel is clear when the received signal strength (energy on the medium) is below a programmable threshold. Mode 2 signals a clear channel when the transceiver does not receive valid IEEE 802.15.4 data. Mode 3 is a combination of the modes 1 and 2. Furthermore, a hysteresis is defined, which has the purpose of avoiding too frequent CCA changes in modes 1 and 3. By signaling a clear

 ISBN: 978-1-61208-308-7

channel only when the energy falls below threshold minus hysteresis, minimal fluctuations do not lead to CCA changes.

A transmission request brings the transceiver to the state TX_CALIBRATE. This phase has a duration of 8 or 12 symbol periods. A parameter named *TX_TURNAROUND* defines which one is used. A duration of 12 symbol periods is compliant with IEEE 802.15.4. Afterwards, preamble and SFD are transmitted (state TX_PREAMBLE), and finally the rest of the frame, taken from a special FIFO memory (state TX_FRAME). After transmitting the frame, the transceiver goes to state RX_CALIBRATE.

B. Integration into ns-3

The simulator ns-3 [4] is a discrete-event network simulator usable for, but not limited to, Internet systems. It is the successor of the well-known ns-2 [3], which is still used in academic research, but is no longer actively developed and has several drawbacks concerning its design.

Figure 2 shows the overall structure of an ns-3 simulation. Applications (for generating and processing traffic), protocol stacks (e. g., UDP / IP), and net devices (which provide Medium Access Control (MAC) functionality and define an interface for the network layer to access a physical device) are installed on nodes. The protocol stack can also be omitted, since applications can be defined in such a way that they communicate directly with net devices. The nodes are then connected by channels. Mobility models can be installed on wireless nodes, determining the positions and movements of these nodes. Besides this, ns-3 provides several loss and delay models, which can be attached to (wireless) channels.

The actual simulation is driven by events, which are delivered to a scheduler. Initially, such events are created by applications running on the nodes. Further events are either created by applications as well or result from the simulation flow (e. g., a send event triggers a receive event at nodes in range).

The part of a simulation system covered by the CC2420 simulation module is marked in Figure 2. By developing the module as a part of ns-3, one can benefit from existing ns-3 components. Existing propagation loss and delay models are used for the CC2420 channel, and predefined mobility models

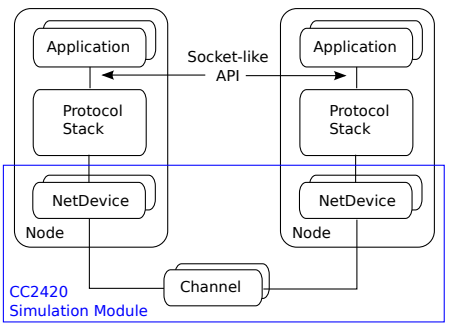


Figure 2. Overall structure of an ns-3 simulation [4].

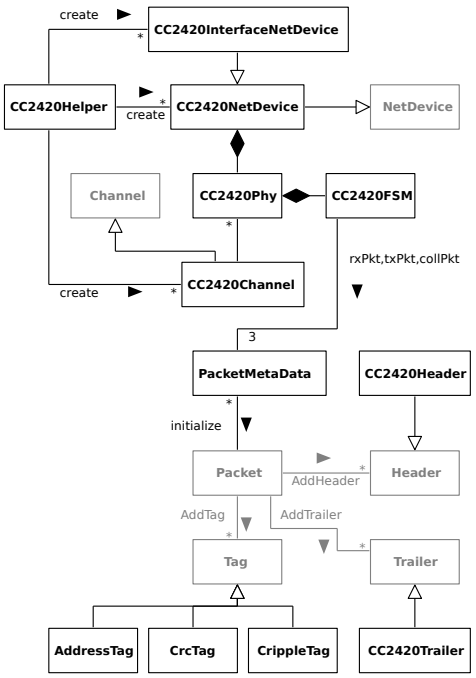


Figure 3. Structure of the CC2420 simulation module.

as well as applications and protocol stacks such as UDP / IP or TCP / IP can be installed on nodes using a CC2420 net device.

A first version of the CC2420 simulation module for ns-3 has been developed in [17] and is based on an existing ns-2 simulation module [5]. Its structure is shown in Figure 3. The classes *NetDevice*, *Channel*, *Packet*, *Tag*, *Header* and *Trailer* and the accordant relations between them are provided by ns-3.

For each node communicating via CC2420, *CC2420Helper* creates a *CC2420NetDevice* or *CC2420InterfaceNetDevice*. Together with the net device, a physical layer is created, which is connected with an existing or newly created *CC2420Channel*.

CC2420NetDevice provides an interface for the simulation module to interact with higher protocol layers. *CC2420InterfaceNetDevice* provides an extended interface, which allows not only sending and receiving of data, but also configuration of the transceiver, etc. (see Section III-C). *CC2420NetDevice* uses a *ReceiveCallback* whose interface is defined in the *NetDevice* class to forward received packets to higher protocol layers. *CC2420InterfaceNetDevice* additionally provides a *MessageCallback*. While a *ReceiveCallback* only allows the reception of regular messages (i. e., ns-3 packets), a *MessageCallback* is used to receive specific messages according to the extended interface.

CC2420Phy holds the configurable parameters of the transceiver (e. g., preamble length, sync word, transmission power, and channel number) and adds an accordant header and trailer (see below) to the ns-3 packets representing the frames of the transceiver. Furthermore, the physical layer controls the *CC2420FSM*, which realizes the state machine described in Section III-A.

 ISBN: 978-1-61208-308-7

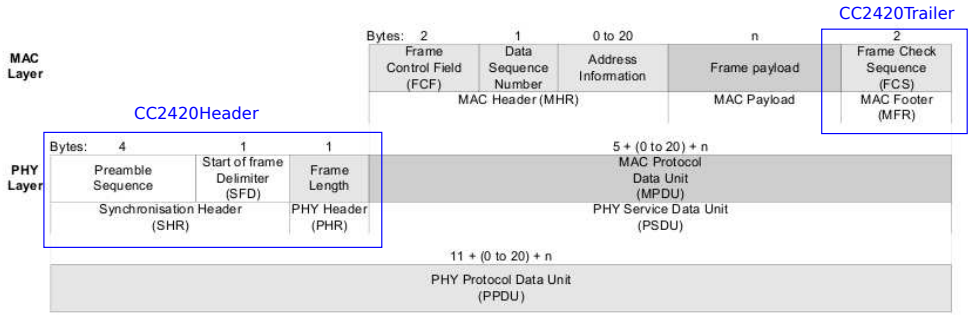


Figure 4. Frame format of the CC2420 transceiver [1].

The state machine used in the simulation module combines the state of the transceiver with the one of the simulated medium. This is the only part which was nearly completely reused from the ns-2 simulation module.

PacketMetaData is a helper class for storing the currently received, currently colliding or currently transmitted frame (represented by an ns-3 packet) together with its received respectively transmitted signal power and its starting time.

CC2420Channel manages the 16 channels of the 2.4 GHz band as subchannels and calculates when a transmitted frame arrives at a node and the signal power. To do this, the mobility models of sender and receiver as well as the transmission signal power are considered. The actual calculations are done by the delay model and the loss model, respectively.

Tags are a possibility to add simulation information to an ns-3 packet without actually extending its length, i. e., its duration on the medium. Source and destination address of a CC2420 frame are stored in an *AddressTag*. The addresses are not encoded in the accordant packet directly, because the CC2420 net device is primarily designed for broadcast transmissions. The actual addressing is done by a higher protocol layer and therefore already encoded in the MAC Protocol Data Unit (MPDU). Therefore, we do not include this information in the packet header, but, for compliance with the ns-3 *NetDevice*, provide it as tag.

CrippleTag is used for marking packets which could not be received correctly, either due to a channel change or because they have an other sync word and are therefore not recognized by the transceiver.

The frame format of the CC2420 transceiver is shown in Figure 4. Preamble Sequence, Start of Frame Delimiter and Frame Length are encapsulated by the class *CC2420Header* and are added to the ns-3 packet. Frame Length denotes the length of the MPDU in bytes; its maximal value is 127, since the highest bit is reserved [1]. Therefore, the MPDU can contain 127 bytes at most. The MAC Header (Frame Control Field, Data Sequence Number and Address Information) is not simulated. Since in the simulation 2 bytes are reserved for the Frame Check Sequence (FCS), the maximal payload size is 125 bytes. The FCS is an additional CRC checksum, which is realized by the class *CC2420Trailer*. It can be configured if this checksum shall be added to the packet. If it is added, the packet is marked with a *CrcTag*, because otherwise, the receiver cannot determine if the checksum is added or not.

The process of successfully sending a message with the *CC2420NetDevice* is as follows: Messages are sent from higher protocol layers in the form of ns-3 packets to the net device, which forwards them to the physical layer. This layer is responsible for registering the send request in the state machine, which causes the physical layer to forward the packet to the channel, when the calibration time has expired. The channel puts the packet to the accordant subchannel and schedules a reception event for all receivers attached to this subchannel by using a timer and an accordant callback. When the timer expires, the scheduler triggers the reception of the packet in the physical layer. A reception request is then sent to the state machine. After reception, the packet is forwarded to the physical layer, which forwards it to the net device. The *ReceiveCallback* is used to deliver it to higher protocol layers.

C. Enhancements

In ns-3, a standard net device only provides the possibility to send and receive packets. But for the CC2420 simulation module, further signals for configuration and information purposes shall be provided. For example, a received packet shall carry its signal strength, CCA changes or the end of a transmission shall be signaled, and changing channel and transmission power shall be supported. In addition, it should be possible to get information about the current configuration of the transceiver. Therefore, we designed the *CC2420InterfaceNetDevice*, which inherits from the standard *CC2420NetDevice* and uses the extended message interface shown in Figure 5.

RawDataMessage provides a generic class to represent payload data. *CC2420Message* provides a unified interface for all messages sent to or received from the *CC2420InterfaceNetDevice*. The messages *CC2420Send*, *CC2420Setup*, *CC2420Config* and *CC2420StatusReq* are sent from upper protocol layers to the simulation module, while *CC2420Recv*, *CC2420Cca*, *CC2420Sending*, *CC2420SendFinished* and *CC2420StatusResp* are sent from the simulation module to upper layers.

 ISBN: 978-1-61208-308-7

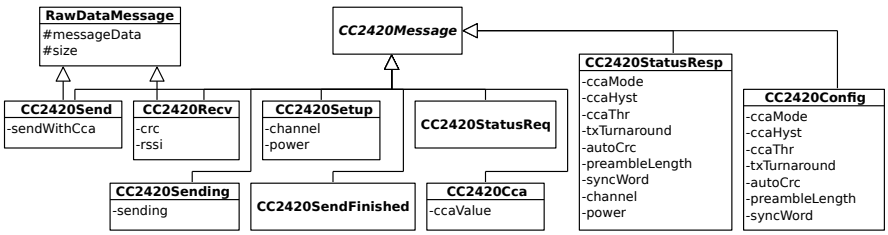


Figure 5. Extended message interface for CC2420 simulation module.

The *CC2420NetDevice* can be used with existing applications and protocol stacks provided by ns-3 without any modifications, since it sends and receives regular ns-3 packets. However, the *CC2420InterfaceNetDevice* requires adaptations in the upper layer(s), i. e., ns-3 applications and ns-3 protocol stacks, because the *CC2420Messages* are to be constructed, and *CC2420Messages* to be received.

The *CC2420Send* message is used to send a packet. It can be configured explicitly if CCA shall be considered or not, i. e., if the medium shall be checked before sending (the standard send method implicitly sends with CCA). *CC2420Recv* contains the received data and additionally provides information on CRC correctness and Received Signal Strength Indicator (RSSI). *CC2420Sending* delivers information about the start of a transmission. The parameter is *true* if the transmission could be started successfully, and *false* if not (e. g., if the medium is busy and sending with CCA is requested). *CC2420SendFinished* has no parameters and is returned when the transceiver has finished a transmission. *CC2420Cca* provides information about the current CCA status and is sent whenever the CCA status changes. *CC2420Setup* and *CC2420Config* are used for configuring the transceiver from higher protocol layers. *CC2420Setup* is used to adjust channel and transmission power, while *CC2420Config* carries values for CCA mode, CCA hysteresis, CCA threshold, TX turnaround, automatic CRC, preamble length and sync word. For the CCA mode, there is no check for valid IEEE 802.15.4 data at the moment. The *CC2420StatusReq* message can be used to get information about the current transceiver configuration. The transceiver module then sends a *CC2420StatusResp* message up, which contains the values of all configuration parameters.

Compared to the CC2420 simulation module for ns-2, we have added the possibility to request the current transceiver configuration. Further, configuration of CCA hysteresis, TX turnaround, automatic CRC, preamble length and sync word are supported, and information on CRC and RSSI is provided. Finally, we have implemented the use of different CCA modes.

IV. Simulator Framework FERAL and Integration of ns-3

In this section, we present the integration of ns-3 and the CC2420 simulation component into FERAL, a simulator framework for the rapid coupling of diverse simulators, such as simulators for Simulink and SDL models.

A. Outline of FERAL

FERAL is a Java-based framework for rapid simulator coupling with the objective to evaluate functional and non-functional requirements of networked systems [8]. A FERAL simulation system consists of a set of simulation components, which are executed by specialized simulators. In particular, existing simulators supporting different kinds of models and targeting different hardware platforms or communication technologies can be used together. Thereby, system components on different levels of abstraction can be simulated, which can, for instance, be applied for early prototyping. One example for this is the use of an SDL simulator together with ns-3 and our CC2420 module, which will be utilized in Section V. Thus, one can simulate existing SDL specifications on a high abstraction level together with a concrete medium model.

The execution of simulation components is controlled by *directors*, which support time-triggered as well as event-triggered semantics. Interaction between simulation components is realized by messages (e. g., event notification).

Three adaptation steps are necessary to build a simulation system with FERAL (see [8]). First, existing simulators, e. g., ns-3, to be used in the simulation system are integrated into FERAL. This is achieved by implementing the *SimulationComponent* control interface of FERAL, which needs to be done only once per simulator. Second, for each integrated simulator and type of simulation component, the FERAL component-specific interface is adapted and implemented. Third, simulation components are instantiated by choosing a simulator integrated into FERAL and by specifying and inserting an accordant behavior or communication model.

B. Integration of ns-3 into FERAL

Since FERAL is written in Java and ns-3 in C++, the integration consists of a Java part and a C++ part, which are connected by the Java Native Interface (JNI). Although ns-3 is an event-based simulator, our component has a time-triggered execution model. This approach was chosen because ns-3 already offers the possibility to execute the simulation for a specified time span. Therefore, no modifications concerning the clock and internal scheduler of ns-3 are necessary.

Figure 6 shows a class diagram of the Java part of the ns-3 simulator component for FERAL. The connection to the C++ part is realized by *NS3Interface* and *NS3Connector*. For each communication medium to be simulated with ns-3, a corre-

 ISBN: 978-1-61208-308-7

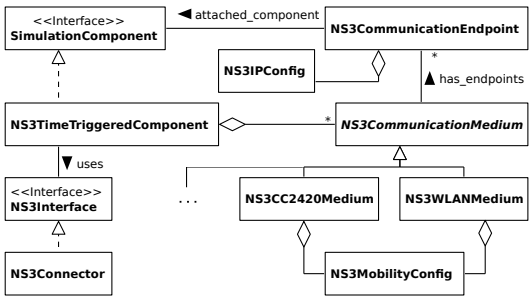


Figure 6. The ns-3 simulator component (Java part).

sponding *NS3CommunicationMedium* is instantiated. For wireless media, an *NS3MobilityConfig* is defined, which determines positions and movements of the nodes attached to this medium. Here, we focus on the CC2420 medium and neglect other media such as Ethernet or WLAN, which are also supported by the simulator component. *NS3CommunicationEndpoints* connect simulation components for functional behavior to the communication medium. On C++ side, these endpoints are represented by special applications.

In a stand-alone ns-3 simulation, applications installed on nodes are used to generate and process traffic. When incorporating ns-3 into FERAL, *virtual applications* are defined, i. e., applications that have no real behavior but rather serve as communication counterparts for the Java endpoints on C++ side. Communication between endpoints and virtual applications is realized via message exchange.

The counterpart of the *NS3CommunicationMedium* is the ns-3 channel, which simulates the actual communication medium. We have introduced a *communication mode* determining the type of communication used for a medium. Currently, the protocols TCP and UDP and communication without using a protocol stack, by sending broadcasts directly via net device are supported as communication modes. When using TCP or UDP, an *NS3IPConfig* must be provided for each endpoint, defining IP address and other parameters of the node. We have implemented one generic message type for all of these modes, which can be used independently of the concrete medium. For each mode, a generic virtual application exists, which transforms the messages to ns-3 packets and sends them according to the communication mode. For TCP and UDP, an accordant protocol stack is installed on the nodes during the initialization. Besides these universal modes, a medium-specific communication mode exists, which is currently only supported for the CC2420 medium.

C. Integration of the CC2420 Simulation Module into FERAL

Besides the integration of ns-3 into FERAL, two further steps are necessary for integrating the CC2420 simulation module. First, a medium class must be provided in order to create a CC2420 medium from a FERAL simulation system. With this medium, it is already possible to use the CC2420 module with the generic message interface described above. To use the CC2420-specific message interface described in Section III-C, the medium-specific communication mode – currently only available for CC2420 – had to be introduced. To use this message interface from FERAL simulation systems, it has to be represented in the framework. Therefore, an (almost) equivalent Java interface has been defined, which mirrors the one from Figure 5. Since communication between the FERAL framework and the ns-3 simulator is done via JNI, accordant code had to be provided to transfer the Java messages to the respective C++ messages.

If the medium-specific communication mode is chosen in combination with the CC2420 medium, a CC2420-specific virtual application is installed on the ns-3 nodes. This application forwards the messages directly to a *CC2420InterfaceNetDevice*, which processes the data and calls the accordant methods. The use of a protocol stack (e. g., IP) is not possible in this case, since communication is done directly via net device.

V. Simulations using the CC2420 Module

In this section, we present results of several simulation experiments, which show that the CC2420 simulation module is fully operational. In particular, we present the use of the CC2420 module in stand-alone ns-3 simulations, and in simulations where ns-3 is a simulator component of the FERAL framework.

A. Stand-alone ns-3 Simulations

In our stand-alone ns-3 simulations, simulation systems consist of two nodes acting as sender and receiver, respectively. An *OnOffApplication*, which sends values during configurable time intervals, is installed on the sender node, while a *PacketSink* is installed on the receiver node. Both applications are provided by ns-3. The structure is shown in Figure 7.

By default, the *OnOffApplication* repeatedly pauses for one second and afterwards sends packets for one second. We start this application at two seconds simulation time and simulate five seconds on the whole, which means that packets are sent in the interval between three and four seconds. By varying application data rate and packet size, we obtain three simulation systems. An excerpt of the first simulation system is shown in Listing 1. UDP sockets are used for sending and receiving (see lines 12 and 17), and the IP address of *PacketSink* is used as destination address for *OnOffApplication* (line 12). The first simulation uses an application data rate of 70 kbps and a packet size of 20 bytes (lines 13 and 14).

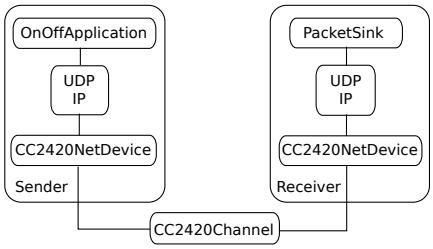


Figure 7. Structure of stand-alone ns-3 simulation systems.

 ISBN: 978-1-61208-308-7

```
NodeContainer nodes; nodes.Create(2);
CC2420Helper cc2420;
NetDeviceContainer devices;
devices = cc2420.Install(nodes, ... ); // CC2420NetDevice
InternetStackHelper stack; stack.Install(nodes);
MobilityHelper mobility;
...
Ipv4AddressHelper addr;
addr.SetBase("10.1.1.0", "255.255.255.0");
Ipv4InterfaceContainer interfaces = addr.Assign(devices);

OnOffHelper onoff ("ns3::UdpSocketFactory",
  InetSocketAddress(interfaces.GetAddress(1), 9));
onoff.SetAttribute("DataRate", StringValue("70kbps"));
onoff.SetAttribute("PacketSize", StringValue("20"));
ApplicationContainer senderApps =
  onoff.Install(nodes.Get(0));

PacketSinkHelper pktSink("ns3::UdpSocketFactory", ... );
ApplicationContainer receiverApps =
  pktSink.Install(nodes.Get(1));
```

Listing 1. Excerpt of simulation system for first simulation.

With these values, all packets reach their destination, which is shown in Listing 2. The total number of bytes sent by *OnOffApplication* equals the number of bytes received by *PacketSink*, which means that all packets have reached their destination.

```
At time 3.00229s on-off application sent 20 bytes to
  10.1.1.2 port 9 total Tx 20 bytes
At time 3.00427s packet sink received 20 bytes from
  10.1.1.1 port 49153 total Rx 20 bytes
...
At time 3.99886s on-off application sent 20 bytes to
  10.1.1.2 port 9 total Tx 8740 bytes
At time 4.00084s packet sink received 20 bytes from
  10.1.1.1 port 49153 total Rx 8740 bytes
```

Listing 2. Output of the first simulation (all packets received).

In the second simulation, an application data rate of 90 kbps instead of 70 kbps is configured. Since UDP and IP headers as well as the CC2420 header and trailer have to be added to the application data rate, and the transceiver calibration time has to be considered, this data rate is too high for the transceiver. Therefore, not all packets can be transmitted, as shown in Listing 3.

The application sends all packets down the protocol stack, but since the CC2420 transceiver can only handle a new transmission request after the current one is finished, some of the packets are discarded by the transceiver. Therefore, we have identified a bottleneck in the system. This behavior can also be identified by additional log outputs of the CC2420 module not shown in the listing.

```
At time 3.00178s on-off application sent 20 bytes to
  10.1.1.2 port 9 total Tx 20 bytes
...
At time 3.99911s on-off application sent 20 bytes to
  10.1.1.2 port 9 total Tx 11240 bytes
At time 3.99932s packet sink received 20 bytes from
  10.1.1.1 port 49153 total Rx 5620 bytes
```

Listing 3. Output of the second simulation.

In the third simulation, the application data rate is set to 70 kbps as in the first one, but the packet size is extended to 150 bytes. This is more than the maximal payload of the transceiver, which is 125 bytes (UDP and IP headers and the CC2420 header and trailer even increase the packet size of the application). Therefore, an IP fragmentation takes place, which means that the IP packet is split into several subpackets to match the Maximum Transmission Unit (MTU) of the *CC2420NetDevice*. However, this does not work with the CC2420 transceiver, because it can handle a new transmission request only after the current one is finished. Because of this, only the first subpacket of each IP packet can be transmitted successfully. Since incomplete IP packets are discarded on network level, *PacketSink* receives no packets at all (see Listing 4).

```
At time 3.01714s on-off application sent 150 bytes to
  10.1.1.2 port 9 total Tx 150 bytes
...
At time 3.99429s on-off application sent 150 bytes to
  10.1.1.2 port 9 total Tx 8700 bytes
```

Listing 4. Output of the third simulation.

Next, we have repeated these simulations with slightly modified *OnOffApplication* and *PacketSink* in order to illustrate the use of the extended CC2420 message interface (see Figure 5). Instead of installing a protocol stack on the nodes, messages are directly forwarded from the application to the net device, which is now a *CC2420InterfaceNetDevice*, and vice versa. This also means that the transmitted packets are smaller, since UDP and IP protocol headers are omitted.

In the modified simulation systems, we have used the extended interface of the CC2420 module to change the channel from the default value 11 to 12 before message exchange is started (in *OnOffApplication* as well as *PacketSink*). A *CC2420StatusReq* message and the corresponding *CC2420StatusResp* message are used to check that the channel change has taken place.

The first of these modified simulations produces nearly the same result as the one with original *OnOffApplication* and *PacketSink*. Since the packets are smaller, they are received slightly earlier. In addition, there are further messages from the *CC2420InterfaceNetDevice*, which are received by the application (see Listing 5). For example, the *CC2420Sending* message with value *true*, which is sent up immediately, indicates a successful transmission start.

```
At time 3.00229s on-off-cc2420 application sent 20 bytes
  total Tx 20 bytes
At time 3.00229s on-off-cc2420 application received
  CC2420Sending message with value true
At time 3.00257s packet sink cc2420 received CC2420Cca
  message with value false
At time 3.00337s on-off-cc2420 application received
  CC2420SendFinished message
At time 3.00337s packet sink cc2420 received 20 bytes
  with CRC=true and RSSI=-67; total Rx 20 bytes
At time 3.00341s packet sink cc2420 received CC2420Cca
  message with value true
...
```

Listing 5. Output of the modified first simulation.

In the second modified simulation, all packets are now successfully transmitted. Since there are no UDP and IP headers, the application data rate of 90 kbps can be handled by the transceiver.

 ISBN: 978-1-61208-308-7

In the third modified simulation, the packet size of 150 bytes now exceeds the MTU size of the net device, since no IP fragmentation takes place. Therefore, no packet is transmitted.

B. Simulations with FERAL and SDL

ITU-T's SDL [9] is a formal specification language designed for distributed and reactive systems. System behavior is defined by extended finite state machines, which are connected through channels and communicate by exchanging signals asynchronously. Channels are also used to connect a system to its environment, e. g., the simulator framework FERAL. The integration of an SDL simulator component into FERAL has been presented in [8]. We use SDL to specify the behavior of nodes in the simulation systems on a high abstraction level, and ns-3 to simulate the medium by means of the CC2420 module.

In the following experiments, simulation systems consist of three nodes, two senders and one receiver, which are specified in SDL and executed by an SDL simulator. These nodes communicate over a wireless medium accessed through a CC2420 simulation module executed by ns-3. The structure is shown in Figure 8. The topology is chosen such that the receiver is positioned between the senders with the same distance to each of them. The extended CC2420 message interface is used for communication with the CC2420 module. The first sender begins transmission at 2,0001 seconds of simulation time and then sends one value every 100 milliseconds. The second sender begins at 3 seconds simulation time and sends one value every 200 milliseconds.

In the first simulation system, the standard configuration of the CC2420 module is applied. In particular, this means that the transceiver transmits at full power. Between 2 and 3 seconds of simulation time, only one sender is active, which means that all signals can be correctly received. Afterwards, every second signal of the first sender collides with a signal of the second sender, which means that only half of the signals of the first sender and none of the signals of the second sender can be received. This behavior is shown in Listing 6.

First, both senders transmit almost simultaneously (lines 2 and 3). Although CCA is used, both transmissions take place, because the calibration time for the first sender has not expired when the second sender begins its transmission. Therefore, the frames collide. The receiver detects that the medium is busy (line 6), but cannot receive a valid frame. Since medium occupancy is only detected by nodes which

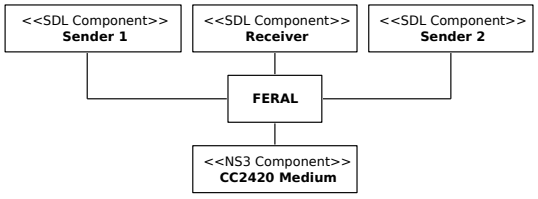


Figure 8. Structure of simulation systems using FERAL.

```
1  ...
2  3,0000: Sender 2: Sent CC2420Send with value 0x20 0x00
     0x00
3  3,0001: Sender 1: Sent CC2420Send with value 0x10 0x00
     0x0a
4  3,0002: Sender 2: Received CC2420Sending with value true
5  3,0003: Sender 1: Received CC2420Sending with value true
6  3,0004: Receiver: Received CC2420Cca with value false
7  3,0007: Sender 2: Received CC2420SendFinished
8  3,0008: Sender 1: Received CC2420SendFinished
9  3,0008: Receiver: Received CC2420Cca with value true
10 3,0010: Sender 2: Received CC2420Cca with value true
11 3,0011: Sender 1: Received CC2420Cca with value true
12 ...
13 3,1001: Sender 1: Sent CC2420Send with value 0x10 0x00
     0x0b
14 3,1003: Sender 1: Received CC2420Sending with value true
15 3,1005: Receiver: Received CC2420Cca with value false
16 3,1006: Sender 2: Received CC2420Cca with value false
17 3,1008: Sender 1: Received CC2420SendFinished
18 3,1008: Sender 2: Received CC2420Cca with value true
19 3,1008: Receiver: Received CC2420Recv with value 0x10
     0x00 0x0b, CRC true, RSSI -67
20 3,1008: Receiver: Received CC2420Cca with value true
21 3,1011: Sender 1: Received CC2420Cca with value true
22 ...
```

Listing 6. Output of the first SDL simulation.

are not in transmission mode, the senders do not detect it here. Since the second sender only transmits every 200 milliseconds, the next frame of the first sender (line 13) can be received successfully (line 19).

In the second simulation system, we use a *CC2420Setup* message to reduce the transmission power of the first sender, while the power of the second sender remains unchanged. Since the distance to the receiver is equal, the signal of the second sender is stronger than the one of the first when arriving at the receiver. This way, a capturing effect can be observed, which means that the reception of the signal from the second sender is not disturbed by the interfering signal of the first sender. This behavior is shown in Listing 7.

```
1  ...
2  3,0000: Sender 2: Sent CC2420Send with value 0x20 0x00
     0x00
3  3,0001: Sender 1: Sent CC2420Send with value 0x10 0x00
     0x0a
4  3,0002: Sender 2: Received CC2420Sending with value true
5  3,0003: Sender 1: Received CC2420Sending with value true
6  3,0004: Receiver: Received CC2420Cca with value false
7  3,0007: Sender 2: Received CC2420SendFinished
8  3,0007: Receiver: Received CC2420Recv with value 0x20
     0x00 0x00, CRC true, RSSI -67
9  3,0008: Sender 1: Received CC2420SendFinished
10 3,0008: Receiver: Received CC2420Cca with value true
11 3,0010: Sender 2: Received CC2420Cca with value true
12 3,0011: Sender 1: Received CC2420Cca with value true
13 ...
14 3,1001: Sender 1: Sent CC2420Send with value 0x10 0x00
     0x0b
15 3,1003: Sender 1: Received CC2420Sending with value true
16 3,1005: Receiver: Received CC2420Cca with value false
17 3,1006: Sender 2: Received CC2420Cca with value false
18 3,1008: Sender 1: Received CC2420SendFinished
19 3,1008: Sender 2: Received CC2420Cca with value true
20 3,1008: Receiver: Received CC2420Recv with value 0x10
     0x00 0x0b, CRC true, RSSI -72
21 3,1008: Receiver: Received CC2420Cca with value true
22 3,1011: Sender 1: Received CC2420Cca with value true
23 ...
```

Listing 7. Output of the second SDL simulation.

As in the first simulation, both senders transmit almost

simultaneously (lines 2 respectively 3). Since the frame of the second sender is stronger, it can be received successfully (line 8). This is only possible because the reception of the frame of the second sender begins before the frame of the first sender, since the transceiver cannot switch from a currently received frame to a stronger one. The next frame of the first sender (line 14) can be received successfully (line 20), since the second sender only sends every 200 milliseconds, which means that there is no colliding frame.

The simulation experiments show that our CC2420 module is fully operational, for stand-alone ns-3 simulations as well as simulations with the framework FERAL. Using simulator components already integrated into FERAL provides additional possibilities for specifying the behavior of nodes compared to stand-alone ns-3 simulations.

VI. Conclusion and Future Work

In this paper, we have developed a medium simulation module for TexasInstruments' CC2420 transceiver. As starting point, we have used an existing module for ns-2. By integrating the new module into ns-3, we can use existing ns-3 protocol stacks and applications to model the behavior of nodes using this transceiver and draw benefit from the active development of ns-3. We have then provided several enhancements for the CC2420 simulation module, which are accessible through an extended interface.

To use the simulation module in combination with other simulators, we have developed an integration into the simulator framework FERAL. Therefore, a general simulator component for integrating ns-3 into FERAL has been provided. This component also supports other ns-3 media, such as Ethernet or WLAN. Next, we have integrated the CC2420 module into this simulator component by providing a Java class for the medium and accordant elements to use the extended CC2420 message interface.

The CC2420 simulation module was first used to simulate stand-alone ns-3 systems. Then, the FERAL simulator component was used to simulate SDL systems which communicate via an ns-3 simulated CC2420 medium. These experiments have shown that the CC2420 simulation module is fully operational in the ns-3 context, and that the integration into FERAL provides additional degrees of freedom especially in the early development stages, where abstract models, e. g., Simulink or SDL models, are used to specify system behavior.

In our future work, we plan to further enhance the simulated state machine. At the moment, interference is not accumulated, which would be desirable for a more precise simulation of collisions. Energy consumption is also an interesting aspect to integrate into the state machine. Besides, we will implement further features of the transceiver, e. g., a check for valid IEEE 802.15.4 data in CCA modes 2 and 3. Furthermore, we will use the CC2420 module to evaluate realistic protocols for mobile ad-hoc networks, e. g., MAC and routing protocols. In addition, we are planning to perform real-world measurements with the CC2420 transceiver, in order to assess how accurate its behavior is simulated by our CC2420 module.

References

[1] Texas Instruments, "CC2420 datasheet," 2013, Revision SWRS041c. [Online]. Available: http://www.ti.com/lit/ds/symlink/cc2420.pdf [Accessed: August, 2013]

[2] IEEE, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). New York, NY, USA: IEEE Computer Society, Oct. 2003. [Online]. Available: http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf [Accessed: August, 2013]

[3] USC Information Sciences Institute, "The Network Simulator – ns-2." [Online]. Available: http://www.isi.edu/nsnam/ns [Accessed: August, 2013]

[4] "The ns-3 Network Simulator," Project Homepage. [Online]. Available: http://www.nsnam.org [Accessed: August, 2013]

[5] T. Kuhn, "Model Driven Development of MacZ – A QoS Medium Access Control Layer for Ambient Intelligence Systems," Ph.D. dissertation, University of Kaiserslautern, 2009.

[6] T. Kuhn and P. Becker, "A Simulator Interconnection Framework for the Accurate Performance Simulation of SDL Models," in System Analysis and Modeling: Language Profiles, 5th International Workshop, SAM 2006, Kaiserslautern, Germany, May 31 – June 2, 2006, Revised Selected Papers, ser. Lecture Notes in Computer Science, R. Gotzhein and R. Reed, Eds., vol. 4320. Springer, 2006, pp. 133–147.

[7] J. L. Font, P. Iñigo, M. Domínguez, J. L. Sevillano, and C. Amaya, "Architecture, design and source code comparison of ns-2 and ns-3 network simulators," in SpringSim, R. M. McGraw, E. S. Imsand, and M. J. Chinni, Eds. SCS/ACM, 2010, pp. 109:1–109:8.

[8] T. Braun, D. Christmann, R. Gotzhein, A. Igel, T. Forster, and T. Kuhn, "Virtual Prototyping with Feral – Adaptation and Application of a Simulator Framework," in The 24th IASTED International Conference on Modelling and Simulation, 2013.

[9] International Telecommunication Union (ITU), "ITU-T Recommendation Z.100 (12/11) – Specification and Description Language – Overview of SDL 2010," 2012. [Online]. Available: http://www.itu.int/rec/T-REC-Z.100-201112-I [Accessed: August, 2013]

[10] C. Suh, J.-E. Joung, and Y.-B. Ko, "New RF Models of the TinyOS Simulator for IEEE 802.15.4 Standard," in WCNC. IEEE, 2007, pp. 2236–2240.

[11] R. de Paz Alberola and D. Pesch, "AvroraZ: Extending Avrora with an IEEE 802.15.4 Compliant Radio Chip Model," in PM2HW2N, B. Caminero, F. Delicado, and R. W. N. Pazzi, Eds. ACM, 2008, pp. 43–50.

[12] H. Joe, J. Lee, D.-K. Woo, P. Mah, and H. Kim, "Demo Abstract: A High-Fidelity Sensor Network Simulator Using Accurate CC2420 Model," in IPSN. ACM, 2009, pp. 429–430.

[13] A. B. Paul, S. Konwar, U. Gogoi, A. Chakraborty, N. Yeshmin, and S. Nandi, "Implementation and Performance Evaluation of AODV in Wireless Mesh Networks using NS-3," in 2nd International Conference on Education Technology and Computer (ICETC) 2010, vol. 5, 2010, pp. V5–298 – V5–303.

[14] H. Narra, Y. Cheng, E. K. Çetinkaya, J. P. Rohrer, and J. P. G. Sterbenz, "Destination-Sequenced Distance Vector (DSDV) Routing Protocol Implementation in ns-3," in SimuTools, J. Liu, F. Quaglia, S. Eidenbenz, and S. Gilmore, Eds. ICST/ACM, 2011, pp. 439–446.

[15] S. Vincent, J. Montavont, and N. Montavont, "Implementation of an IPv6 Stack for NS-3," in 2nd International Workshop on NS-2 (WNS2 2008), October 2008.

[16] N. Baldo and M. Miozzo, "Spectrum-aware Channel and PHY layer modeling for ns3," in VALUETOOLS, G. Stea, J. Mairesse, and J. Mendes, Eds. ACM, 2009, pp. 2:1–2:8.

[17] R. Groh, "Contributions to the Integration of CC2420 and SDL into ns-3," Bachelor Thesis (in German), University of Kaiserslautern, Computer Science Department, 2012.

 ISBN: 978-1-61208-308-7

Physical Layer Simulation of Large Distributed Automation Systems in SPICE

Patrick Diekhake, Eckehard Schnieder
Institute for Traffic Safety and Automation Engineering
Technische Universität Braunschweig
Braunschweig, Germany
e-mail: diekhake@iva.ing.tu-bs.de, e.schnieder@tu-bs.de

Abstract—**In this paper, we propose an analysis of the physical layer of large distributed automation systems based on simulation in SPICE. For large systems, changes of the physical topology or minor modifications of the physical layer hardware of a bus node result in much more increased influences to the signal integrity compared to smaller system architectures. The simulation provides references for further designs of the physical layer hardware of a bus node and an analysis of its behavior in different large topology configurations with up to 1000 bus nodes in a network with a total length of up to 1000 meters.**

Keywords- large distributed automation systems; physical layer simulation; signal integrity; fieldbus simulation

I. INTRODUCTION

Due to the increase in performance and the cost reduction of micro-processors, the decentralized approach of automation systems is used more and more in the last years. Furthermore, current developments of automation systems focus on distributed systems to handle modular extensibility and to avoid isolated applications. The decentralized approach suggests a connection of each sensor and actuator to a bus system via a bus coupler. As a result a large distributed automation system is generated, of which behavior is dependent on a high number of influence factors, based on the complex structure of the transmission channel system and the large number of bus participants.

Current research on physical layer simulations in works mainly in the field of automotive bus systems ([1]-[6]) or focused on lower scale case studies ([7], [8]). In this paper the focus lies on more complex bus system topologies, for example hardware architectures in decentralized building automation systems, which contain more than 1000 bus couplers which are networked in a transmission channel system of up to 1 km without a repeater. Changes of the physical topology or minor modifications of the physical layer hardware of a bus node, such as stub lines, mismatches or unsuitable dimensioning, could cause a negative influence to the signal integrity. A simulation based analysis is essential for this kind of large distributed automation systems.

Based on the fieldbus system SmallCAN ([9] and [10]), developed at the Institut for traffic safety and automation engineering, Technische Universität Braunschweig, the signal integrity for a large distributed automation system is validated by a physical layer simulation in SPICE.

Therefore, the main components models of the mentioned automation system and their connection to a large distributed system are described in section 2 of this paper.

In section 3 the main requirements for a sufficient signal integrity behavior in SmallCAN are introduced.

In section 4 the results are shown after simulation. Next to specific complex topology configurations with up to 1000 bus participants, the influence of modifications of the physical layer hardware of a bus node is also be analyzed.

II. Simulation model

To simulate the data transmission system for SmallCAN, three main components of the data transmission system are modeled in SPICE:

- energy supply unit.
- physical layer hardware of a bus coupler.
- transmission channel system.

The energy supply unit provides a recessive signal level of 24 V for the data line and a constant current source which impresses a current of 250 *mA* in case of a forced dominant signal level. The developed model *ENERGY_SUPPLY_UNIT* supports three ports for ground, input power supply and data line.

The dominant signal level is caused by short-circuiting the data line to the ground, executed by a MOSFET-Transistor of an active bus coupler. The developed model *BUS_COUPLER* represents a bus coupler with the logical input port for transmitting data, the data line of the bus, ground and the logical output port for the receiving data.

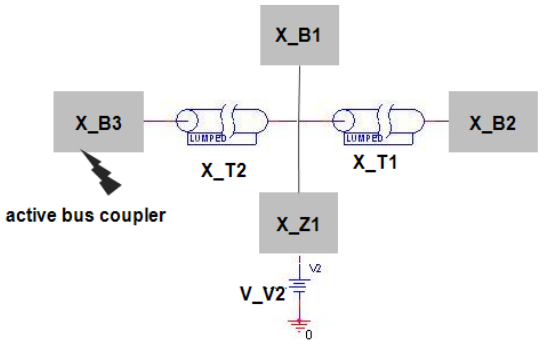
The transmission channel system of SmallCAN considers the electrical parameters for a typical phone cable:

- capacitance per unit $C' = 55\ \frac{\mathrm{pF}}{\mathrm{m}}$.
- resistance per unit $R' = 0{,}05\ \frac{\Omega}{m}$.
- inductance per unit $L' = 0{,}075\ \frac{uH}{m}$.

For the transmission channel system the SPICE model TLUMP128 is used. The individual models of the components are connected to a whole system model of the specific hardware architecture, described by a SPICE net list. The physical topology can be chosen free, as long as the total cable length is lower than 1000 m.

 ISBN: 978-1-61208-308-7

Figure 1 shows a minimal topology with 3 bus couplers at a maximum distance of 3 *m* and the accordance net list. Two phone cables with 1 *m* and 2 *m* are connected between the energy supply unit X_Z1 and the bus couplers X_B2 and X_B3. A pulse, generated by the voltage source V_V3, controls the logical input port of bus coupler X_B3 to trigger the dominant signal level on the bus. The energy supply unit feeds in a current of 250 *mA* in the data line in the direction of the active bus coupler X_B3.




```
V_V2  1      0                                  30Vdc
X_Z1  0      2    1                             ENERGY_SUPPLY_UNIT
X_B1  TX_1     2     0          RX_1            BUS_COUPLER
X_T1    3   2    GND_1  TLUMP128  PARAMS: R=0.05 L=0.075u C=55p  LEN=2
X_B2  TX_2     3      GND_1   RX_2              BUS_COUPLER
X_T2    4   2    GND_2  TLUMP128  PARAMS: R=0.05 L=0.075u C=55p  LEN=1
X_B3  EVENT  4      GND_2   RX_3               BUS_COUPLER
V_V3 EVENT  GND_2  PULSE 0V 5V 1000u 10n 10n 0.1m 0.198m
```

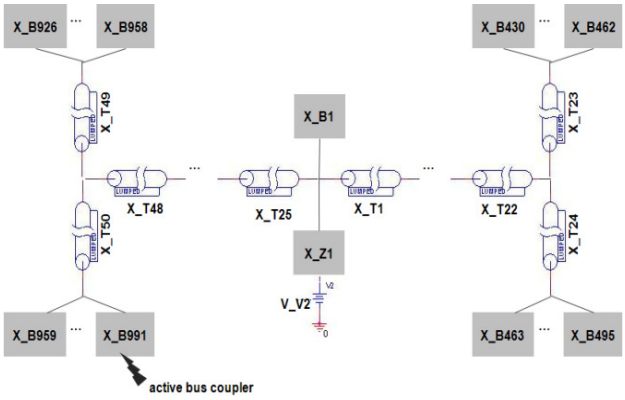
Fig. 1. Minimal scale hardware architecture and SPICE-net list.

In Figure 2 an example of a maximal periodically structured topology with 991 bus couplers of a total cable length of 1 *km* and a section of its net lists are depicted. The energy supply unit X_Z1 is placed in the middle of the system, from where two main lines were extended. Each following stub line has a length of 20 m. At each end of the stub line a network, composed of 33 bus couplers, is located.

III. Requirements

After short-circuiting, the following signal curve depends on signal integrity properties, such as reflection behavior and voltage changes. The following requirements have to be fulfilled for each topology configuration to guarantee a sufficient signal integrity behavior:

- The voltage drop between the data line and the ground has to be lower than 7 V / 500 m, which guarantees the recognition of the dominant signal by the receiver hardware.
- After the settling time and line delay time of 11,9 us the level signal on the data line has to be lower than the threshold voltage of 14 V during the falling edge and has to be overrun it during the rising edge, to ensure a stable signal trace before the signal sampling starts.




```
V_V2  1      0                                  30Vdc
X_Z1  0      2    1                             ENERGY_SUPPLY_UNIT
X_B1  XX_1  2     0   20001                     BUS_COUPLER
X_T1    3   2      T1_0   TLUMP128   PARAMS: R=0.05 L=0.075u C=55p  LEN=20
…
X_T22   22   22   T23_0  TLUMP128   PARAMS: R=0.05  L=0.075u C=55p  LEN=20
X_T23   24   23   T24_0  TLUMP128   PARAMS: R=0.05  L=0.075u C=55p  LEN=20
    X_B430   XX_430    24  T24_0  20430          BUS_COUPLER
    …
    X_B462   XX_462    24   T24_0  20462         BUS_COUPLER
X_T25   25   23   T25_0  TLUMP128   PARAMS: R=0.05  L=0.075u C=55p  LEN=20
    X_B463   XX_463    25   T25_0  20463         BUS_COUPLER
    …
    X_B495   XX_495   25   T25_0  20495          BUS_COUPLER

X_T25   26   2    T26_0  TLUMP128   PARAMS: R=0.05 L=0.075u C=55p  LEN=20
…
X_T48   50   48   T48_0  TLUMP128   PARAMS:  R=0.05  L=0.075u  C=55p LEN=20
X_T49   51   50   T49_0  TLUMP128   PARAMS:  R=0.05  L=0.075u  C=55p LEN=20
    X_B926 XX_926   51   T49_0   20926          BUS_COUPLER
    …
    X_B958 XX_958   51   T49_0   20958          BUS_COUPLER
X_T50    52   50  T50_0  TLUMP128  PARAMS:  R=0.05  L=0.075u  C=55p LEN=20
   X_B959  XX_959   52   T50_0   20959          BUS_COUPLER
   …
   X_B991    EVENT   52  T50_0   20991          BUS_COUPLER
V_V3 EVENT  T50_0 PULSE 0V 5V 1000u 10n 10n 0.1m 0.198m
```

Fig. 2. Large scale hardware architecture and section of the SPICE-net list.

IV. Simulation Results

In Figure 3 (a),(b), the simulation results are depicted for the mentioned large distributed topology. For comparison an ideal trace for the minimal system with three bus couplers by a total cable length of 3 m is also depicted in Figure 3 (c),(d). The voltage between the data line and data ground is measured near the active bus coupler X_B991(cyan), near the energy supply unit at bus coupler X_B1 (red) and far from the active bus coupler at the bus coupler X_B495 (blue).

The simulation results show that the voltage drop between data line and ground is lower than the permitted value of 7 *V*. Due to the influence of the line capacity the signal trace results in a much flattened curve in contrast to the signal curve for the minimal system. It can be also seen that during the falling edge more signal levels are formed,

 ISBN: 978-1-61208-308-7

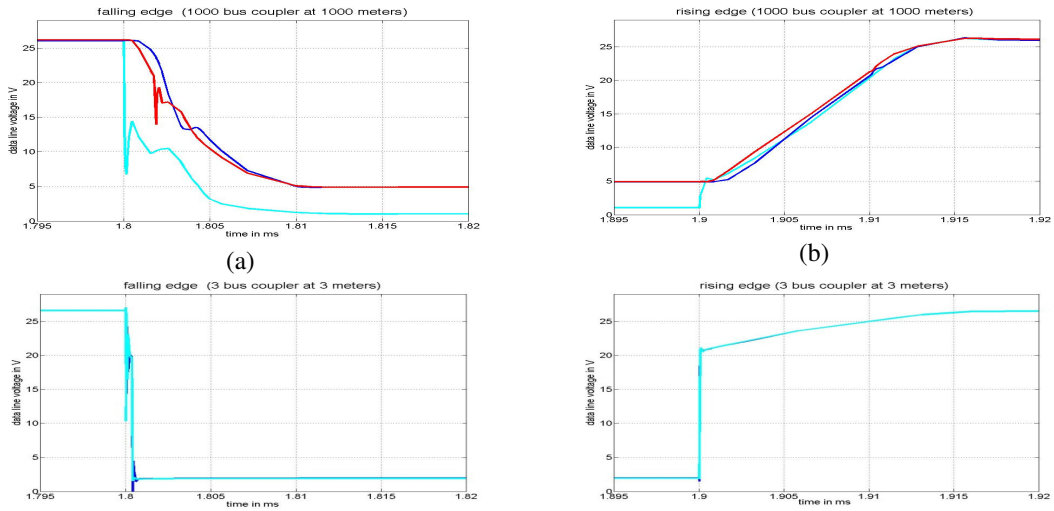


Fig. 3. Simulation results for a large and minimal distributed topology.

due to the reflection behavior. After a line delay time of nearly 2,5 *us* and the settling time the signal is fall below the threshold voltage of 14 *V* within the permitted time of 11,8 *us*.

The trace of the steep edges, affiliated under minimal topology conditions, cannot be kept for extended topology configurations, due to the resulting flattened curve. Furthermore, for smaller system topologies the steep edges result in excessive emitted interferences. Therefore, some modifications of the physical hardware of the bus coupler are implemented to avoid the steep edges during signal level changes. By a delayed triggering of the output transistor, the current flow in the data line is impressed under controlled conditions to avoid the steep edges.

In Figure 4, the simulation results after the hardware modifications of the transmitter are depicted. Due to the controlled triggering of the output transistor, the signal behavior for the large distributed topology resembles more the signal behavior for the minimal system, compared to the simulation results of Figure 3. The mentioned signal integrity requirements are fulfilled and the reflection behavior is reduced. In summary it could be assumed, that for different topology configurations the signal trace on the data line is more predictable after the mentioned hardware modifications.

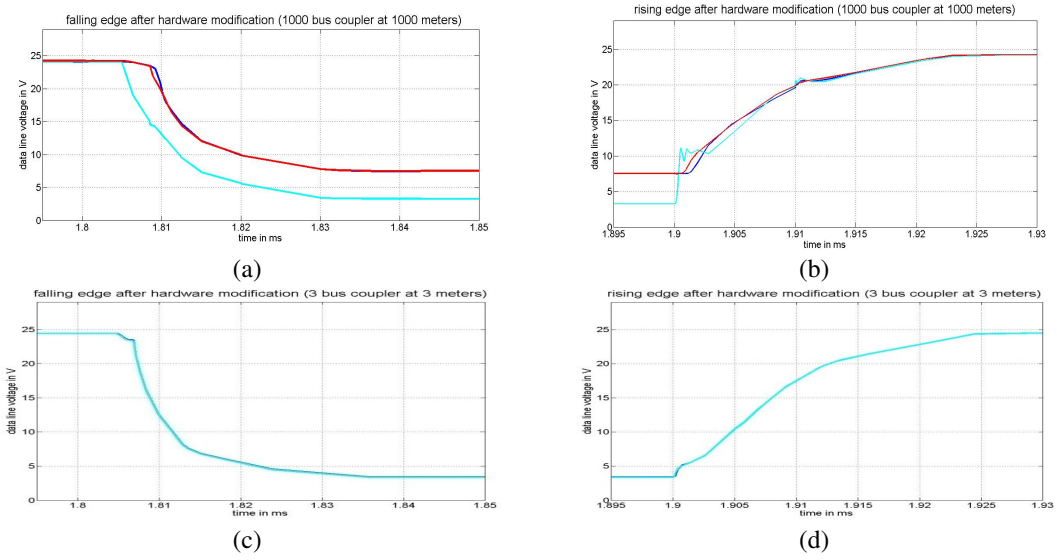


Fig. 4. Simulation results for a large and minimal distributed topology after hardware modification.

 ISBN: 978-1-61208-308-7

V. CONCLUSION AND OUTLOOK

The effort for a measurement based analysis of the physical layer of large distributed automation systems is extremely high and not practical. Therefore, a simulation based analysis for this kind of automation systems is proposed in this contribution. The main component models of an automation system and theirs variable connection by a topology net are described. The simulation results for different case studies show various signal behaviors, dependent on different bus topologies and modifications of the transceiver hardware. Variable parameters allow the analysis of signal integrity for different conditions. Generally, statements about the signal integrity for complex automation systems can be yielded by this simulation.

In further works, an optimized signal behavior for all conditions could be found by an automated modification of parameters, simulation execution and an analysis of the results. Monte Carlo simulations, offered by SPICE Tools, could be used for that.

REFERENCES

[1] H. Günther, S. Frei, T. Wenzel, "Simulation Methods for Signal Integrity of Automotive Bus Systems," Proc. IEEE Symp. Asia-Pacific International Symposium on Electromagnetic Compatibility (APEMC), 2010, pp. 512-515, doi: 10.1109/APEMC.2010.5475788.

[2] M. Krammer et al., "Exploration of the FlexRay Signal Integrity using a Combined Prototyping and Simulation Approach," Proc. IEEE Symp., Design and Diagnostics of Electronic Circuits and Systems (DDECS), 2010, pp. 111-116, doi: 10.1109/DDECS.2010.5491806.

[3] J. Nan, W. Men, "Research on Reflection of CAN Signal in Transmission Line," Proc. IEEE Symp. Intelligent Control and Automation (WCICA), , 2008, pp. 7707-7710, doi: 10.1109/WCICA.2008.4594128.

[4] T. Nguyen, J. Haase, G. Pelz, "Sensitivity analysis of passive CAN bus components to investigate signal integrity of CAN network physical layer," Behavioral Modeling and Simulation Workshop (BMAS), 2008, pp. 55 – 60, doi: 10.1109/BMAS.2008.4751240.

[5] W. Prodanov, M. Valle, R,Buzas, "A Controller Area Network Bus Transceiver Behavioral Model for Network Design and Simulation," Industrial Electronics, 2009, pp. 3762 – 3771, doi: 10.1109/TIE.2009.2025298.

[6] C. Muller, M.Valle, "System verification of flexray communication networks through behavioral simulations," Behavioral Modeling and Simulation Conference (BMAS), 2010, pp. 1 – 6, doi: 10.1109/BMAS.2010.6156589.

[7] W.Ruomin, J. Denoulet, S. Feruglio, F. Vallette, P. Garda, ".High Level Modeling of Signal Integrity in Field Bus Communication with SystemC-AMS," Electronics, Circuits and Systems (ICECS), 2012, pp. 889 – 892, doi: 10.1109/ICECS.2012.6463519.

[8] K. Kraft, "Simulation von CAN-Bus-Komponenten und -Übertragungsnetzen," unpublished.

[9] H. Schrom, T. Michaels, S. Stein, R. Ernst, "SmallCAN - A Reliable, Low-Power and Low-Cost Distributed Embedded System for Energy Efficient Building Automation," Energy2011, May 2011, pp. 13-18.

[10] P. Diekhake, J. Liu, E. Schnieder, "Buslast- und Schaltungssimulation zur Validierung des optimierte Feldbussystems SmallCAN bei maximaler Auslastung," KommA, September 2011, unpublished.

 ISBN: 978-1-61208-308-7

A New Distributed Parallel Event-driven Timing Simulation for ECO Design Changes

Seiyang Yang, Doohwan Kwak, Jaehoon Han,
Dept. of Computer Eng.,
Pusan National University
Busan, Korea
e-mail: syyang@pusan.ac.kr

Namdo Kim,
Infra Design Technology,
Samsung Electronics Co.,
Kyunggi-do, Korea
e-mail: nd.kim@samsung.com

***Abstract* - Prediction-based distributed parallel event-driven hardware description language simulation on multi-core computing platforms is a new promising approach to boost simulation performance. Traditional distributed parallel event-driven simulation methods suffer heavy synchronization and communication overhead for transferring the signal data among local simulators, which could easily nullify most of the expected simulation speed-up from parallelization. In our approach, the signal data to be transferred is predicted first in each local simulation independently. No synchronization and communication incurs when the prediction succeeds, and the actual signal data transfer with synchronization and communication among the local simulators occurs only when the prediction fails. Therefore, as far as prediction accuracy remains high, the high simulation speed-up from the parallelization can be anticipated from the approach. In this paper, we have proposed the prediction-based distributed parallel event-driven timing simulation for a series of design changes in typical ECO flow. We also have performed the preliminary experimentation in actual design changes, obtained high prediction accuracy with real designs from industry and achieved significant speed-up gain from the proposed parallelization.**

Keywords—distributed parallel simulation; synchronization; communication; partitioning; simulator; simulation; verification; EDA

I. INTRODUCTION

Simulation has still remained the most popular verification method in chip designs because of ease of use, low cost, 100% signal controllability and observability, etc. Specifically, event-driven Hardware Description Language (HDL) simulation is the most common technique used for functional and timing simulations [1]. However, event-driven simulation suffers from very low performance for complex design objects because of its inherently sequential nature. In chip designs, this has gotten much worse in gate-level simulation than Register Transfer Level (RTL) simulation because the number of simulation objects to be dealt with is much larger at gate-level than at RTL. But, the use of gate-level functional or timing simulation is still quite active and even increasing nowadays for many important reasons [16][17]. Some of them include verification requirement for designs having many asynchronous clocks, limitation of static functional and timing verification methods such as equivalence checking and static timing analysis, variability of deep sub-micron processing technology, etc. Therefore, event-driven HDL simulation is heavily used for both functional and timing verification. Usually, once the bug is found and fixed after a simulation run, another new simulation run is required with a new HDL code or netlist to ensure that the bug is correctly removed and no new bug is accidently brought. This process is iterated until the designers or verification engineers believe no more bugs exist in the design. In this sense, simulation in the design flow is a highly repeated process before and after a series of continuous design changes.

Distributed parallel event-driven HDL simulation has been proposed to alleviate the low performance of sequential simulation [2][3][4]. Unfortunately, it has been not successful because of: i) difficulty in design partitioning; ii) heavy synchronization and communication overhead among modules imposed by the distributed environment, especially in gate-level timing simulation; and iii) load balancing among the distributed simulation jobs.

This paper consists of following; first we briefly mention the previous work and motivation in Section II, and explain our unique and noble approach to distributed parallel simulation in Section III. In next main section, we propose the prediction-based distributed parallel event-driven timing simulation for a series of design changes in the typical ECO (Engineering Change Order) flow, and claim that the performance of gate-level timing simulation could be greatly improved from the proposed approach. In Section V, we have performed some preliminary experiments with real SOC (System On Chip) designs from industry for demonstrating the expected benefit, followed by the conclusion and future work in Section VI.

II. PREVIOUS WORK AND MOTIVATION

The area of distributed parallel simulation is rich in literature. All known works concern traditional distributed parallel simulation, which is based on physical partitioning of the design into multiple sub-designs, assigned to individual local simulators. This simulation concept has been known since late 1980s as Parallel Discrete Event Simulation (PDES) [9]. The main issues in PDES are partitioning, synchronization and granularity. There are basically two types of

 ISBN: 978-1-61208-308-7

synchronization methods in distributed parallel simulation: *conservative* (lockstep based) and *optimistic* (rollback based). These two types differ in the way modules of the partitioned design communicate during simulation for synchronization. Their performance varies with the design and partition strategy, but usually the optimistic method is faster [9][10][11][12]. Most recently, Chatterjee [13] and Zhu [14] proposed the distributed parallel event-driven gate level simulation using general purpose GPUs (Graphic Processing Units). However, it could only handle gate level zero-delay simulation. It is known that it is not effective for the gate level timing simulation [15]. In conclusion, these methods are not practical, do not scale, and have performances depending heavily on optimal partitioning [18].

Recently, some Electronic Design Automation (EDA) vendors have introduced parallel event-driven HDL simulators for multi-cores [5][6][7][8]. Parallel HDL simulation with multi-core technology looks more promising than the original distributed parallel HDL simulation. In multi-core parallel simulation, (inter-module) communication can be accomplished by a straightforward fast memory read/write. However, the expected speed-up was observed only for a special class of designs, such as BIST (Built in Self Test) logic just for gate-level zero-delay simulation, and the speed-up curve quickly saturates with the number of available cores. The problem becomes particularly difficult for large number of cores, which quickly increases the global communication and synchronization overhead among partitioned sub-designs. It mainly comes from the fact that the difficulty of partitioning for distributed parallel simulation lies in simultaneously considering the reduction of the synchronization and communication overhead and load balancing among distributed simulation jobs. Due to the huge design size, much heavier synchronization and communication overhead, etc., the problem even gets much worse for gate-level timing simulation, where the stronger demand for high simulation performance exists, but its speed is even slower than that of gate-level zero-delay simulation at least by one order of magnitude.

In summary, the success of traditional distributed parallel event-driven simulation on multi-core strongly depends on such "ideal" partitioning, which itself is a known intractable problem and therefore is impossible to apply to complex industrial large designs [18]. This is the main reason that multi-core parallel event-driven HDL simulators are not popular in the design community these days although there has been a great demand for increasing gate-level simulation performance on multi-core platforms.

III. PREDICTION-BASED DISTRIBUTED PARALLEL EVENT-DRIVEN HDL SIMULATION

Yang [3][4] had proposed a new promising approach to boost up the simulation performance, prediction-based distributed parallel event-driven HDL simulation on multi-core computing platforms. This approach is based on *predicting* input and output stimulus that need to be applied to module(s) in each local simulation (We will call each of individual simulation in distributed parallel simulation local simulation). How to accurately predict input and output values is explained in the next section.

The predicted input values are stored in local memory and applied to the input ports of a local module assigned to a given simulator. Then, the actual output values at the output ports of that module are compared on-the-fly with the predicted output values, also stored in a local memory. Figure 1 shows an example design consisting of two modules dependent on each other inputs. Simulating the modules in parallel requires predicting inputs for each of the two sub-modules.

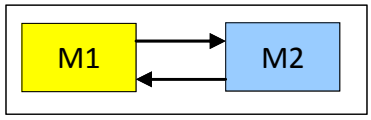


Figure 1: An example design with two dependent sub-modules

Figure 2 shows two sub-modules being simulated in parallel on two cores. Each sub-module uses predicted inputs by default, while its actual outputs are compared against the predicted outputs (stored earlier in local memory). A multiplexer at each sub-module selects between the predicted inputs and the actual inputs. Note that, when both sub-modules access their actual inputs from the other sub-module, synchronization and communication overhead incurs.

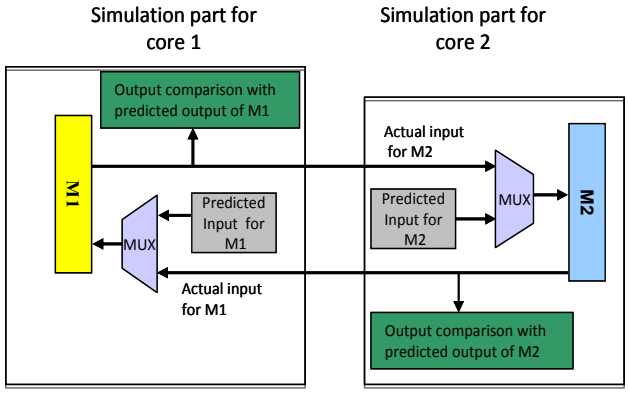


Figure 2: Conceptual diagram of prediction-based distributed parallel event-driven HDL simulation

As long as the prediction is correct, communication and synchronization between two local simulations is completely eliminated. We call this phase of simulation the prediction phase. Only when the prediction fails, the actual input values, coming from the other local simulation, are used in simulation; we call this phase of simulation the actual phase. When prediction fails, each local simulation must roll back to the nearest checkpoint and is restarted from that point. This is possible by generating *checkpoint,* i.e., saving simulation state

 ISBN: 978-1-61208-308-7

or design state, during the simulation in the prediction phase. Note that, when parallel simulation enters the actual phase, it should try to return to the prediction phase as early as possible to attain the maximum speed-up. This is done by continuously comparing the actual outputs of all local simulations with their predicted outputs and counting the number of matches on-the-fly. If the number of matches exceeds a predetermined value, the simulation is switched back to the prediction phase. Therefore, it is obvious that prediction accuracy is the most critical factor in this approach. Prediction accuracy near 100% will give almost linear speed-up even when the number of processor cores increases.

IV. PREDICTION-BASED DISTRIBUTED PARALLEL EVENT-DRIVEN TIMING SIMULATION FOR ECO DESIGN CHANGES

As being mentioned in the previous section, having an accurate prediction data is imperative in the prediction-based distributed parallel event-driven HDL simulation. In [3], Yang had proposed one way to obtain the accurate prediction data, i.e., from the earlier simulation with the higher abstraction model. For example, if the prediction-based distributed parallel simulation is for gate-level simulation with a gate-level netlist, the prediction data could be gathered from the earlier RTL functional simulation with the higher abstraction model, i.e., Verilog RTL design, from which the netlist is going to be synthesized.

In this paper, we propose another way to obtain the accurate prediction data in the design flow. When we apply this prediction-based distributed parallel timing simulation at gate level for verifying design changes in ECO flow, our idea is to get it from signal dump from the earlier simulation before design change. For example, for doing the distributed parallel simulation with the example design in Figure 1 on two cores, all input and output ports of two sub-modules M1 and M2 are registered for signal dump. While simulating the design before design change, signal dumping is performed for those input and output ports of sub-module M1 and M2, and saved as the prediction data for later simulation after design change. Let's assume that a bug in sub-module M1 is revealed from the 1st simulation, and its HDL source or netlist is modified for fixing it. Then, prediction-based distributed parallel timing simulation deployed for the 2nd simulation after design change will use this saved signal dump on the input and output ports of sub-modules M1 and M2 as the prediction data.

As mentioned earlier, simulation is the highly iterated activities before and after a series of continuous design changes in the entire design phase. The application of distributed parallel HDL timing simulation to verify design changes in ECO flow could result in a profound benefit for reducing the total simulation time of all simulation runs, due to this repeating nature. Again, the key factor for boosting the simulation performance from the prediction-based distributed parallel simulation is the prediction accuracy. Intuitively, higher prediction accuracy for the prediction-based distributed parallel simulation in ECO flow is expected from less design change. For example, very high prediction accuracy could be expected from timing-only variant design changes keeping same functionality. But, there are other factors to consider. First, although there is a design change inside a module, it may only internally affect the module, but not at its output port and beyond. In other words, the effect of the change could not be propagated to any of its output port. If this is the case, the prediction accuracy of signal dump from the simulation before design change is 100%. This is the best scenario for prediction-based distributed parallel simulation. Second, a design change inside a module could affect any of its output port, but not its input port. This is the case when there is no feedback connection from its output to its input, possibly through some other module(s), gates(s), etc. For this case, the prediction accuracy of input signal dump from the simulation before a design change is still 100%. But that of output signal dump is not 100%. Third, a design change inside a module could affect both its output and input port, when there is some feedback connection from its output to its input. For this, the prediction accuracy of signal dump from the simulation before design change is not 100%. This might be the worst scenario, and its prediction accuracy cannot be 100%. But, if its accuracy is still high, we could achieve the substantial speed-up from the distributed parallel simulation.

Most design change(s) in ECO flow at the back-end design stage seldom changes the functional behavior of the corresponding sub-module globally in the entire simulation scenario. In fact, many ECO design changes are adjustments only for timing, e.g. buffer substitution, insertion, or removal, functionally equivalent but timing different cell substitution, cell re-placement, re-routing, etc. All of these ECO design change examples do not alter the functional behavior at all. Therefore, we intuitively know that the prediction accuracy for prediction-based distributed parallel event-driven timing simulation could be quite high in typical ECO flow.

Besides reducing communication and synchronization overhead, another important factor to consider in distributed parallel simulation is load balancing. In fact, significant speed-up from distributed parallel simulation is only possible by satisfying these two conditions simultaneously. Another benefit of applying the prediction-based distributed parallel event-driven timing simulation for ECO design changes is that a good load balancing could be easily known from a simulation run before design change. That is, while the prediction data is being collected during the simulation run before design change, the simulation load profiling for major design sub-blocks in design is also carried out. In this case, the prediction data being collected is the signal dump of all input and output ports of those major design sub-blocks.

As any ECO design change, especially timing variant but function invariant design change seldom or never alter the simulation behavior drastically. The simulation load profiles

 ISBN: 978-1-61208-308-7

before and after design change should be pretty much the same. Therefore, as the simulation load of a new simulation run after design change could also be accurately predicted from that of the simulation run before design change, a good load balancing as well as the low communication and synchronization overhead could be anticipated from the prediction-based distributed parallel event-driven timing simulation in ECO flow.

In series of successive ECO design changes, all remaining simulation runs except the 1st simulation run can utilize the signal dump of all input and output ports of the major design sub-blocks as the accurate prediction data for prediction-based distributed parallel simulation. To execute the 1st gate-level timing simulation run also in parallel fashion, the prediction data could be brought from the gate-level functional simulation, i.e. gate-level zero-delay simulation. Therefore, the entire timing simulation runs in ECO flow could be run in parallel, and we could expect that the total simulation time for entire ECO design changes is greatly reduced.

In the next section, we have performed some preliminary experiment to justify our claims with real designs from industry.

V. PRELIMINARY EXPERIMENTATION

In this section, we provide some interesting preliminary experimental results for measuring the prediction accuracy and estimated speed-up from the proposed approach in ECO design change with real industrial designs. The ECO design changes are confined to timing variant but functional invariant. The Verilog simulator we used for the experiment is one of leading commercial Verilog simulators. The first test design in Table I is a BIST design. Here, a design change is a cell replacement for adjusting the timing while keeping the original functionality.

For parallel simulation, the design is partitioned into 5 pieces according to the simulation activities. To do this, the profiling feature in the commercial Verilog simulator had been used.

First, we have measured the prediction accuracy from the ECO design change. The measuring procedure is the following; i) while running the simulation before the corresponding ECO design change with an original design, the prediction data is collected by dumping the signal values on all input and output ports of all modules at the partition boundary, ii) the ECO design change is performed, iii) while running the simulation after the corresponding ECO design change with a modified design, the actual data is collected by dumping the signal values on all input and output ports of same modules at the partition boundary, and iv) comparing the prediction data with the actual data. Note that this comparison should be event-by-event basis. The resulting prediction accuracy in this case is 99.9%. This means that during 99.9 % of the total simulation time each local simulation can be run independently without incurring any communication and synchronization. The communication and synchronization among five local simulations is required only for 0.1 % of the total simulation time. Other additional factors to be considered are checkpoint overhead, and rollback and restart overhead. In this design, only a single rollback is needed. By considering all these, the expected speed-up from the proposed method in the specific ECO design change is 5.12. It is pretty surprising and almost too good to believe.

TABLE I. EXPERIMENTAL RESULT 1

Design Name	Prediction Accuracy (%)	# of Partitions	Expected Speed-up from the proposed method
BIST	99.9	5	5.12

At first glance, it seems this 5.12x speed-up is by no means possible from any distributed parallel simulation with 5 partitions. This is true for any traditional distributed parallel simulation methods that always require communication and synchronization during the entire simulation time. Then, how about for the proposed prediction-based distributed parallel simulation method? The answer is it is possible. The reason is the following. When each local simulation is run independently in the prediction mode, it is possible that even the speed of the slowest local simulation is greater than *(the speed of the non-parallel original simulation)/(# of partitions)*. In this design, the non-parallel original (single core) gate-level timing simulation takes 10,916 sec (wall clock time). In the proposed prediction-based distributed parallel simulation, the slowest local simulation running in the prediction mode only takes 2,130 sec., which is shorter than a fifth of 10,916 sec. We think this comes from the fact that the smaller design avoids virtual memory trashing, which leads to low CPU utilization and degrades the system performance. Note that the unavoidable heavy communication and synchronization nullify this potential benefit in the traditional distributed parallel simulation. We believe that this is a very interesting and important finding that largely differentiates the proposed approach from others.

TABLE II. EXPERIMENTAL RESULT 2

Design Name	Prediction Accuracy (%)	# of Partitions	Expected Speed-up from the proposed method
Mobile AP	99.6	8	Not Available

The second test design is a state of art mobile Application Processor (AP) design from the industry. Again, a design change is a cell replacement for adjusting the timing while keeping the original functionality. For parallel simulation, the design is partitioned into 8 pieces according to the simulation activities. However, due to the security reason, unfortunately we have only measured the prediction accuracy from the ECO design change for the second design. It is shown in Table II.

 ISBN: 978-1-61208-308-7

Like the first design, the very high prediction accuracy has been observed for the second design, too.

From the experiment with two real designs from the industry, we strongly believe that our prediction-based distributed parallel HDL simulation is very effective for boosting the simulation performance at least for timing-only variant ECO design changes.

VI. CONCLUSION AND FUTURE WORK

HDL simulation is a very iterated process before and after a series of design changes. Prediction-based distributed parallel HDL simulation is a new promising approach to parallelize the simulation. Its effectiveness heavily relies on the prediction accuracy. As near 100% accuracy can eliminate most of synchronization and communication overhead, the speed of parallel simulation could significantly be increased. In this paper, we have applied the prediction-based distributed parallel event-driven HDL timing simulation for ECO design changes in chip designs.

We have experimentally shown that in the timing-only variant design changes the accurate prediction data for the distributed parallel simulation after design change could be obtained from the earlier simulation before design change, and contribute to the large decrease of communication and synchronization overhead. Therefore, almost linear speed-up from the parallelization could be anticipated. In the future, we would like to extend the application scope of this approach to the function variant design changes as well as timing-only variant design changes. Our final goal is to implement the prediction-based distributed parallel event-driven HDL simulation method on commercial Verilog simulators, and it is under investigation.

REFERENCES

[1] T. Anderson and R. Bhagat, “Tackling Functional Verification for Virtual Components,” ISD Magazine, November 2000, pp. 26.

[2] P. Rashinkar, P. Paterson, and L. Singh, System-on-a-Chip Verification: methodology and techniques, Springer, 2002.

[3] D. Kim, M. Ciesielski, and S. Yang, “A new distributed event-driven gate-level HDL simulation by accurate prediction,” Design and Test Europe (DATE 2011), March 2011, pp. 547-550.

[4] T. B. Ahmad, N. Kim, B. Min, A. Kalia, M. Ciesielski, and S. Yang, "Scalable Parallel Event-driven HDL Simulation for Multi-Cores," Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design, (SMACD 2012), Sept. 2012, pp. 217-220.

[5] IUS Multicore datasheet, Cadence Design Systems (http://www.cadence.com).

[6] VCS Multicore datasheet, Synopsys (http://www.synopsys.com) .

[7] SimCluster datasheet, Avery Design Automation (http://www.averydesign.com)

[8] MP-Sim datasheet, Axiom Design Automation (http://www.axiomda.com)

[9] R.M. Fujimoto, “Parallel Discrete Event Simulation,” Communication of the ACM, Vol. 33, No. 10, Oct. 1990, pp. 30-53.

[10] A. Gafni, “Rollback Mechanisms for Optimistic Distributed Simulation Systems,” SCS Multiconference on Distributed Simulation, Vol. 3, July 1988, pp. 61-67.

[11] R.M. Fujimoto, “Time Warp on a Shared Memory Multiprocessor,” Transactions of the Society for Computer Simulation, Vol. 6, No. 3, July 1989, pp. 211-239.

[12] D.M. Nicol, “Principles of Conservative Parallel Simulation,” Proceedings. of the 28th Winter Simulation Conference, 1996, pp. 128–135.

[13] D. Chatterjee, A. DeOrio, and V. Bertacco, “Event- driven gate-level simulation with general purpose GPUs,” Proceedings. of Design Automation Conference (DAC09), June 2009, pp. 557-562.

[14] Y. Zhu, B. Wang, and Y. Deng, “Massively Parallel Logic Simulation with GPUs,” Article No. 29, ACM Trans. On Design Automation of Electronic Systems, June 2011.

[15] K. Chang and C. Browy, “Parallel Logic Simulation: Myth or Reality?” Computer, Vol. 45, No. 4, April 2012, pp. 67-73.

[16] Gate-level Simulation Methodology, Whitepaper, Cadence Design Systems (www.cadence.com), 2013

[17] VerificationBlog (http://whatisverification.blogspot.com/2011/06/gate-level-simulations-necessary-evil.html)

[18] L. Li and C. Tropper, “A design-driven partitioning algorithm for distributed Verilog simulation,” in Proc. 20th International Workshop on Principles of Advanced and Distributed Simulation (PADS), June 2007, pp. 211–218.

 ISBN: 978-1-61208-308-7

GRIND: An Generic Interface for Coupling Power Grid Simulators with Traffic, Communication and Application Simulation Tools

David Chuang
Shanghai Jiao Tong University
SEIEE
Shanghai, China
unknownmight@sjtu.edu.cn

Björn Schünemann, David Rieck
Fraunhofer FOKUS
ASCT
Berlin, Germany
{bjoern.schuenemann, david.rieck}@fokus.fraunhofer.de

Ilja Radusch
Technische Universität Berlin
OKS / DCAITI
Berlin, Germany
ilja.radusch@dcaiti.com

***Abstract*—The prospective penetration of the electric vehicle fleet will bring about certain repercussions due to their high demand in power. Simulations are, therefore, of importance for making estimations for assessing the impact of the incoming electric vehicle fleet on the power systems and to predict some cost specific values. To analyse the working conditions of power grids, power system simulators are not to be dispensed with. However, to have a complete picture of a charging scenario involving electric vehicles, further aspects should be preferably observed, e.g., traffic, communication and application aspects. So far, no sophisticated tool exists that incorporates the further simulation aspects for a comprehensive investigation of electric mobility. To address this issue, this paper proposes a concept for enabling the coupling of power system simulators with simulators of other domains. The concept is described in form of a specification called Grid Analysis Interface Definitions (GRIND). As a proof of concept, the V2X Simulation Runtime Infrastructure (VSimRTI) and the electrical power system simulator OpenDSS are coupled following the proposed GRIND specification.**

***Keywords*—VSimRTI; Simulation Tools; Electric Mobility.**

I. INTRODUCTION

Recent research and development is continuously striving to create innovation that improves the standard in driving and minimize hazardous situations on the roads. To address the issues about the local CO_2 emissions, the vehicular industry is shifting toward focusing on manufacturing electric vehicles [1]. However, the upcoming plug-in electric vehicle (PHEV) fleet might have certain negative impact on the power grid due to their high power demand. In order to reduce risks and repercussions of a prospective penetration of a fleet, a good foresight must be obtained before the roll-outs take place. Especially testing is crucial for making accurate estimations to assess the impact of the incoming PHEV fleet on the power systems and to predict some cost specific values. One option to do field testing vehicular set-ups might require non trivial budgets, and they are rigid and non-flexible. The other option is to resort to simulations. There exists several simulation tools for power system analysis. Power system simulations alone, however, do not suffice in order to conduct analysis on charging patterns of PHEV's. To have a complete picture of the elaborate happenings, a traffic simulator for modelling vehicular traffic, a communication simulator to facilitate an information exchange among traffic participants and infrastructure units, and an application simulator for emulating in-vehicle and mobile applications should be incorporated into the simulation environment. So far, no simulation environment is available, which provides a sophisticated modelling of all these aspects.

To amend the described shortage, this paper proposes a concept for interconnecting power system simulators with simulators from other domains. The work is inspired by TraCI[2], "a technique for interlinking road traffic and network simulators" to facilitate research on the VANET domain. As a proof of concept, a concrete implementation will be done by coupling the open source load flow simulator OpenDSS [3] with the powerful simulation framework VSimRTI [4] that enables the coupling of simulators of different research domains. VSimRTI is a promising candidate since it already couples existing traffic, communication, and application simulators.

This paper is structured as follows: In Section II, relevant work will be presented including the simulation architecture VSimRTI. Concepts for realizing the coupling process and made design decisions follow in Section III. Moreover, implementation details are given. Finally, the proof of concept is introduced in Section IV, and a conclusion is given in Section V.

II. BACKGROUND

A. Simulation Couplings

A notable work of high relevance is the **Tra**ffic **C**ontrol **I**nterface (TraCI) [2]. TraCI is an API designed to "interlink road traffic and network simulators", it is a generic protocol specification that allows external programs to control the microscopic and macroscopic vehicle behaviour in a traffic simulation from outside. To design the concept, the authors recognized the fact that vehicular behaviour can be broken down into atomic operations called "mobility primitives". Each one of those mobility primitives were used to set a basis for constructing a message. An important feature of the TraCI interface is that it was made generic and is, therefore, neutral to simulation specific details. This feature allows any vehicular simulation tool to become a TraCI server and any program to be the client. TraCI also adheres to a server and client architecture, which allows it to be platform-independent and

 ISBN: 978-1-61208-308-7

it also allows the communication to take place over different machines.

Another work of relevance is presented by Andersson, Elofsson, Galus et al., who conducted a wide range of research in the PHEV domain [5], [6]. They proposed a framework that couples the energy hub concept with an extended version of MATSim in order to investigate their ideas. However, the coupling does not allow trivial replacement of the particular tools since the framework was not designed in a generic manner.

B. VSimRTI

VSimRTI [4] is a generalized framework for the coupling of different simulators, each for a particular domain, following an ambassador concept inspired by some fundamental concepts of the High Level Architecture (HLA). All management tasks, such as synchronization, interaction and lifecycle management are handled completely by VSimRTI. Several optimization techniques, such as optimistic synchronization, are implemented. The generic VSimRTI interfaces allow an easy integration and exchange of simulators. Consequently, the deployment of simulators is enabled for each particular domain. For immediate use, a set of simulators is already coupled with VSimRTI: the traffic simulators VISSIM and SUMO; the communication simulators ns-3, OMNeT++, JiST/SWANS, and a cellular communication simulator; a Java-based application simulator; and several visualization and analysis tools. VSimRTI is a promising candidate for the objectives of this work. Therefore, it is chosen as the underlying system for coupling power system simulators with simulators from other domains.

III. Concept of Realization

A. Requirements

This paper proposes the **Gr**id Analysis **In**terface **D**efinitions (GRIND), a specification for the flexible coupling of power system simulators with simulators of other domains. For the realization of GRIND, the following requirements were defined:

- GRIND is to allow interactions between a power grid simulator and simulators of different domains. Interactions occur during the runtime of a simulation. That means, the coupled simulation tools can retrieve and change the state of the power grid simulator during the runtime of a simulation.
- GRIND is specified in a generic way so that it can be used with an arbitrary power grid simulator. Furthermore, its interfaces are to be flexible enough for the coupling of simulation tools of different domains.
- GRIND is to enable distributed simulations, i.e. the coupled simulation tools can run on different machines and operating systems.

In the following sections, the concept is explained, which has been developed to fulfil these requirements.

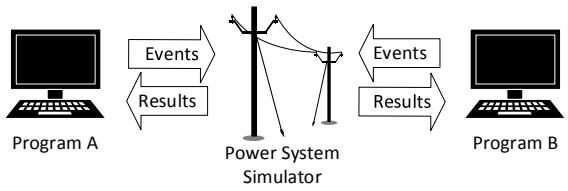


Fig. 1. Interaction between programs and a power system simulator

B. Approach

The aim of GRIND is to provide a specification, which enables developers to couple power system simulators with other tools. Since TraCI [2] follows a similar approach for the VANET domain – to couple traffic simulators with other simulation tools, some concepts of GRIND are inspired by TraCI concepts.

Even if all existing power system simulators provide similar services, each of them has its own way to model its inner working. Some simulators have similar ways for modelling the grid elements, while others use different calculation methods. However, since the simulation models of all simulators are based on the same theories and principles for load flow analyses, some more abstract "information" can be identified that apply to all power system simulators. For example, most power system simulators include loads and generators as part of a circuit albeit in different formats. The high level notion of a load, therefore, is applicable to any power system simulator without having to regard how it is internally modelled. In addition to this abstract data, several events exist, which change the current state of a grid. Typical examples are the addition or the removal of a load triggering the increase/decrease of the power consumption. Analogous to the fact that the same high level information is processed by any power system simulator, events will likewise be independent of the used power system simulator. Regarding the interactions, the power system simulator acts as a provider for data related to the grid. In terms of events, they can be triggered by both – the grid simulator and the external system. Consequently, the exchange of information and events have to be standardized in a way that both sides understand the communication.

A typical work flow is as follows. The power system simulator and another simulator with an interest in grid related data initiate their communication. The grid simulator internally performs any calculations needed to solve the state of the power grid. Once in a while, the external simulator sends queries or update changes to the power system simulator, which are used by the power system simulator to update the state of the power grid. This work flow is depicted in Figure 1.

C. GRIND Server and Client

In order to establish the communication, a channel has to be set up for a bi-directional message exchange. For that matter, simulators have to be extended by the needed functionality to be compatible with GRIND. However, simulators come in different strengths with respect to extensibility. Some simulators can be augmented with ease, while others do not allow expanding practices. To cover every possible

 ISBN: 978-1-61208-308-7

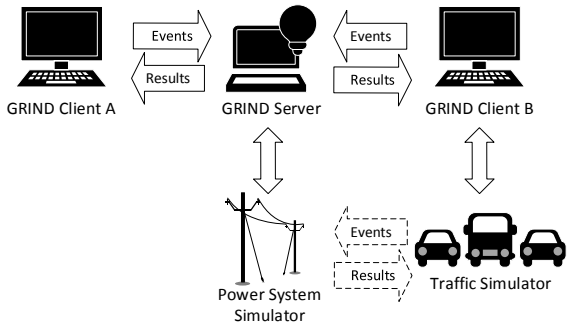


Fig. 2. Interaction between programs and a power system simulator according to GRIND

setup, GRIND implements a server and client architecture. The communication is realized by network sockets, which guarantees a platform independent use of GRIND. Moreover, client and server do not need to be installed on the same machine. According to different setups, the GRIND server or clients can act as either a middleware or an extension of a particular simulator following the GRIND specification. A middleware is needed if a simulator is self-contained and does not support extensions. If a tool is extensible, the interfaces for GRIND can be integrated into its system without the need of a middleware.

The proposed architecture is depicted in Figure 2. The dashed arrows indicates, which parties having a conversation. To realize this conversation in the described generic way, the power system simulator establishes a connection to the GRIND Server. Then, the GRIND Server connects the GRIND Client B, which is coupled to the traffic simulator. In this way, the power system simulator and the traffic simulator can interact with each other.

D. GRIND Messages

To have an interaction between simulators, it has to be defined what kind of information can be exchanged. Both parties have to be able to interpret the received information – i.e. the server and the client have to "speak the same language". For that purpose, GRIND specifies a set of information that is grouped into discrete units. Such an information unit is termed "message" where one message encompasses several related information. The content of the different message types is defined by GRIND in a way that all needed information can be transferred by the available pool of message types. To transfer an information, the suitable message type is chosen, the information is encapsulated there, and, then, the message is sent. Since the message type is known, the other side can interpret the received message.

Most existing power simulators are able to share certain features in common. These features are used to infer information that are universally applicable. Using these features as a foundation similar to the concepts of "mobility primitives" introduced by TraCI [2], actions can be identified. These actions are used to define the message types of the GRIND protocol. Since the pool of messages is to cover an area as wide as possible, not every message is universally applicable. In other words, some power system simulators, providing fewer functionality than others, disregard message types they cannot process.

The following paragraphs give a brief introduction of the message types defined by GRIND. Since the space of this paper is limited, the messages are described on a higher level.

1) NewFile: Most existing power system simulators support the setup of a circuit by loading configuration files albeit in different formats. In the case where these files are not stored on the server side, the client can use a NewFile message indicating the incoming transmission of an actual file.

2) CreateCircuit: Certain power system simulators model an internal circuit within their system prior calculation. This message can be used to prompt the power system simulator to use any existing resource for constructing a circuit.

3) Topology: In order to avoid a redundant parsing, a client is not aware of the structure of a circuit. This message type contains those pieces of grid data, which are to be sent to a client.

4) ChangeLoad, ChangeGeneration: Although the topology remains static, the load dispatch is highly variable. By these message types, common changes in the load configuration are transferred by the client to the server to update the grid when necessary.

5) NewLoad, NewGenerator, RemoveLoad, RemoveGenerator: Loads or generators can be added or removed from the system with help of these message types. However, it is not very common to remove a generator in a running system.

6) SolveGrid: The most important service provided by a power system simulator is to perform a power flow calculation. Since most tools do not perform this action automatically, this message type requests the power system to solve the grid using its current parameters.

7) GridResults: Not every client needs the same parameters from a solved power grid. For example, some clients might only need the total line values, while other clients could require the detail state of the entire topology in one minute steps. Therefore, it is not efficient to specify each parameter as one single message. Instead, an aggregate message that is freely adjustable is defined by GRIND. The detailed content of the message can be specified by the developer according to the need of data of a particular simulator. This message is sent as a response to the SolveGrid message.

E. Addressing Scheme

Each power system simulator models its circuits in a different manner. Some simulators define the entire circuit within matrices while other more sophisticated ones virtually model the elements. In order to address individual elements, the developer has to come to an agreement in form of an addressing scheme. For instance, if a load is saved within a cell in a matrix on the third row and fifth column, this load can be uniquely addressed by using "3,5" as identifier. Consequently, this identifier can be included in a ChangeLoad message whenever the load "3,5" has to be increased in power. In contrast, other simulators, for example OpenDSS, name the buses and do not need such a naming process. Instead, they simply indicate the literal name in a ChangeLoad message.

 ISBN: 978-1-61208-308-7

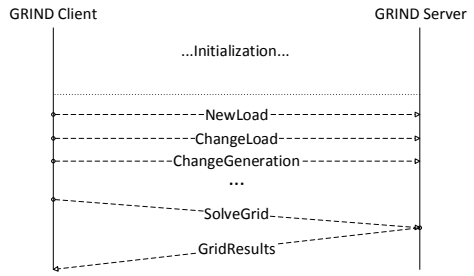


Fig. 3. An example of a message flow following the GRIND protocol

F. Protocol

Since TCP is the underlaying communication protocol, it can be assumed that all messages do arrive in order. Most messages do not require a direct response from the server side, however, they aim to trigger an activity the server is to start. A common scenario is to involve the server to change the load parameters on a bus by a ChangeLoad message or remove an entire generator capacity from a bus with help of a RemoveGeneration message. However, one particular message does require a reply from the GRIND server: When the client asks for results of the power flow calculation using the SolveGrid message. An example of a message flow is depicted in Figure 3.

IV. Proof of Concept

A. Implementation using OpenDSS and VSimRTI

For realizing the proof of concept, the power system simulator OpenDSS and the simulation architecture VSimRTI were coupled following the GRIND specification. OpenDSS was selected because of its richness in features and its well-designed interfaces. The advantage of VSimRTI is that it is already coupled with several traffic, communication, and application simulators. Thus, a coupling of OpenDSS and VSimRTI creates a simulation environment, which can cover a wide range of different simulation aspects. In the planned simulations, the traffic simulator SUMO, the communication simulator JiST/SWANS, and the VSimRTI application simulator are integrated in the VSimRTI simulation setup – additionally to the power system simulator OpenDSS.

B. Scenario

For the planned simulations, a test scenario is set up where the electric grid is presented by the IEEE 30 test feed [7]. The selected area of the simulation is the City of Roanoke (USA). The overall electric grid, including changes induced by charging processes, is modelled by OpenDSS. Roanoke map data from OpenStreetMap[8] are used to model the road network. The vehicular traffic is generated by SUMO. An in-vehicle application is implemented, which guides the driver to an unused charging station and controls the charging processes. The information exchange among vehicles and infrastructure units is simulated by JiST/SWANS.

C. Aim of the Proof of Concept

Additionally to the investigation of performance issues like scalability and simulation speed, the proof of concept is to demonstrate that the coupling of OpenDSS and VSimRTI following the proposed GRIND specification is a promising approach to enable comprehensive simulations of electric mobility scenarios. The different aspects electric grid, vehicular traffic, information exchange among traffic participants and infrastructure units, and emulation of in-vehicle and mobile applications can be modelled by this solution. Since all these aspects influence each other during the runtime of a simulation, a dynamic coupling is necessary, which allows interactions among the simulators during a simulation run. The proof of concept shall illustrate that the realized coupling fulfils these requirements and, thus, enables more detailed investigations of electric mobility and its impacts.

V. Conclusion

This paper proposes a concept for the flexible coupling of power system simulators with simulators of other domains. As the result, simulation tools from different domains can be linked to an arbitrary grid simulator. This is particularly helpful for comprehensive investigations of electric mobility where the influences and interactions of power grid, vehicular traffic, communication, and in-vehicle applications are to be considered, e.g., in cooperative ITS. The generic server-client architecture of the proposed GRIND specification allows cross-platform compatibility and platform independence. The implemented coupling of the power system simulator OpenDSS and the simulation architecture VSimRTI follows the GRIND specification and, hence, creates a simulation environment, which enables a comprehensive assessment of novel electric mobility solutions. In the next step, the introduced proof of concept will be simulated to demonstrate the effectiveness and potency of this work.

Acknowledgment

The presented research work is part of the eMERGE project financially supported by the German Federal Ministry of Transport, Building and Urban Development (BMVBS).

References

[1] E. Knipping and M. Duvall, “Environmental assessment of plug-in hybrid electric vehicles volume 2: United states air quality analysis based on aeo-2006 assumptions for 2030,” *Electric Power Research Institute*, 2007.

[2] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, “TraCI: an interface for coupling road traffic and network simulators,” in *Proceedings of the 11th communications and networking simulation symposium*. ACM, 2008, pp. 155–163.

[3] R. Dugan and T. McDermott, “An open source platform for collaborating on smart grid research,” in *Power and Energy Society General Meeting, 2011 IEEE*, 2011, pp. 1–7.

[4] B. Schünemann, “V2x simulation runtime infrastructure vsimrti: An assessment tool to design smart traffic management systems,” *Computer Networks*, vol. 55, pp. 3189–3198, October 2011.

[5] M. Galus, R. Waraich, M. Balmer, G. Andersson *et al.*, “A framework for investigating the impact of phevs,” 2009.

[6] M. Geidl, G. Koeppel, P. Favre-Perrod, B. Klockl, G. Andersson, and K. Frohlich, “Energy hubs for the future,” *Power and Energy Magazine, IEEE*, vol. 5, no. 1, pp. 24–30, 2007.

[7] R. Christie, “Power systems test case archive,” *Electrical Engineering dept., University of Washington*, 2000.

[8] M. Haklay and P. Weber, “Openstreetmap: User-generated street maps,” *Pervasive Computing, IEEE*, vol. 7, no. 4, pp. 12–18, 2008.

 ISBN: 978-1-61208-308-7

Personalizing Thermal Comfort in a Prototype Indoor Space

Sotirios D Kotsopoulos, Federico Casalegno
School of Humanities Arts and Social Sciences
Massachusetts Institute of Technology
Cambridge, Massachusetts, USA
e-mail: skots@mit.edu, casalegno@mit.edu

Antoine Cuenin
Department of Energy Systems Engineering
Ecole Nationale Supérieure des Mines
Nantes, France
e-mail: antoine.cuenin@gmail.com

***Abstract*—This paper presents an experimental method for monitoring and regulating thermal comfort at the interior of a house. It presents a function that was developed in MATLAB, aiming to compute the Predicted Mean Vote (PMV) and Predicted Percentage of Dissatisfied (PPD) indexes. Furthermore, an improvement of the MATLAB function was developed in the form of a Simulink simulation by using fuzzy logic. Although the method, at its current stage of development, cannot compute a PMV index value in the way it was intended, it does compute the operative temperature, allowing the residents to ascertain whether or not thermal comfort can be established in a particular indoor area.**

Keywords-thermal comfort; simulation; fuzzy logic.

I. INTRODUCTION

Thermal comfort is of paramount importance in ensuring fine living conditions in indoor spaces. This paper presents an experimental method for monitoring and regulating thermal comfort at the interior of a prototype house that is at the final stage of construction in Trento, N. Italy. The theory of thermal comfort is used, based on the work of P. O. Fanger who introduced two indexes, the Predicted Mean Vote (PMV) and Predicted Percentage of Dissatisfied (PPD), in order to quantify the parameter of thermal comfort [1]. This study presents a function that was developed in MATLAB [2] to compute thermal comfort based on simulation. The function computes the PMV and PPD indexes at the same time and displays the results on a graph. Furthermore, an improvement of the MATLAB function was developed in the form of a SIMULINK simulation [3] by using fuzzy logic [4]. Although the overall system at its current stage cannot compute a PMV index value in the way it was intended at the beginning of the research, it does compute the operative temperature, thus allowing the residents to ascertain whether or not thermal comfort can be established in a particular indoor house area.

Numerous attempts to quantify thermal comfort have been made since the 1970's, when the Danish scientist P.O Fanger established a thermal comfort theory. Although Fanger published his first paper on the subject with the intention to establish a higher standard for the use of Heating Ventilating and Air Conditioning (HVAC) systems, the theory of thermal comfort was neglected by the environmental management industry. The approach of traditional environmental management systems towards indoor comfort relies mostly on tracking the fluctuation of the interior temperature and humidity levels and reactively adjusting the operation of the HVAC system. This approach is not different from using simple thermostatic control. Furthermore, traditional environmental management systems regulate the thermal conditions of an interior at a building scale and individual considerations are neglected.

The design philosophy of the prototype house in Trento aims at a personalized environmental management approach [5], [6]. The south façade of the prototype is a programmable solar wall, forming a grid of electro-active windows, the modifications of which enable the precise adjustment of light, heat, view, and air in the interior. On a hot summer day, the electrochromic layer of a number of windowpanes can be set to its minimum solar transmittance value to protect the interior from direct sun exposure. On a cold winter day, it can be set to its maximum solar transmittance value to expose the interior to the winter sun. One of the project's objectives is to minimize the use of electricity. The residents determine the desired comfort levels and their schedules, and a central control system minimizes the consumption of electricity while guaranteeing that the comfort levels are maintained at all times. Given that the Trento prototype was envisioned to provide an environment that remains adaptable to individual human needs, it was only natural to attempt developing an environmental management system that would acknowledge the thermal comfort level of individual users.

After providing a brief overview of related work to this research, a MATLAB function is presented able to calculate the thermal comfort levels at the interior of the prototype house. After presenting simulations with the MATLAB function, tools involving fuzzy logic are integrated in the method of computing the PMV and PPD values, aiming to provide greater flexibility and accuracy in the calculations. The system that is presented in the paper combines two parts operating in parallel, a *personal-dependent model* and an *environmental model*. A series of calculations of the PMV index are presented next, with the motivation to prove that the proposed system in its current state, while not fully operational, it is still competent. It simulates PMV and in real life application it can supply a temperature that is in close proximity to the one that achieves thermal comfort. Finally, the paper ends with the conclusions and with suggestions for further research.

 ISBN: 978-1-61208-308-7

II. BACKGROUND

Thermal comfort accounts for numerous parameters that influence decisively a person's feeling of comfort in an indoor space. Multiple atmospheric parameters have to be considered if one is to determine the thermal comfort in a room, namely, *air temperature*, *mean radiant temperature*, *air velocity*, and *relative humidity*. *Mean radiant temperature* is the uniform temperature of the surrounding surfaces, which will result in the same heat exchange by radiation from a person as in the environment. *Air velocity* is a parameter that is relevant to the consideration of the heat loss by convection. *Air velocity* is directly related to the amount of energy exchanged in this physical process. *Relative humidity* is an important aspect of the global heat loss process. If the air is dry the *relative humidity* is low, and perspiration increases to help keep the body cool. A high level of *relative humidity* will prevent perspiration, and it will have the opposite effect of a low *relative humidity*.

Thermal comfort is based on the notion of the human body heat balance. The human body produces energy and heat that is required for its operation by burning calories. When there is a balance between heat production and heat loss, the body's temperature is at 37 C° and Fanger's equation is satisfied:

$$S = M \pm W \pm R \pm C \pm K - E - RES = 0. \quad (1)$$

where:

S = Heat Storage
M = Metabolism
W = External work
R = Heat exchange by radiation
C = Heat exchange by convection
K = Heat exchange by conduction
E = Heat exchange by evaporation
RES = Heat exchange by respiration

Metabolism represents the amount of heat released inside the human body once calories have been burned. *Heat loss by evaporation* accounts for the body heat loss by the diffusion of water through the skin. This water uses body heat to evaporate, thus contributing to heat loss. *Heat loss through clothing* accounts for the heat exchange of body heat with the external environment by diffusion, through clothes. Clothes worn by a person affect the amount of the exchanged heat. Experiments with thermal mannequins in thermal chambers had determined the thermal insulation provided by clothes [7]. *Heat loss by radiation* amounts to the heat exchange of the body with its surroundings through radiation. *Heat loss by convection* corresponds to one of the major modes of heat transfer. The heat is "carried away" by the airflow. The amount of heat exchanged depends on the air velocity. The *air velocity* determines whether or not the convection is "free" or "forced".

To calculate the degree of thermal comfort within a group of people, Fanger devised the PMV index. The PMV is calculated by Fanger's comfort equation (1). Each time the equation is satisfied under specific conditions, a large group of people experience thermal neutrality, or thermal comfort.

Table I presents the seven-level scale that the PMV index employs to capture the mean thermal sensation vote of a group of people.

TABLE I. PMV SEVEN-LEVEL SCALE

Scale	***Feeling***
-3	Cold
-2	Cool
-1	Slightly cool
0	Neutral
1	Slightly warm
2	Warm
3	Hot

The PMV model indicates the level of thermal comfort for a group of healthy adults. It is a statistical estimation of the mean thermal comfort and hence some individuals may still feel uncomfortable even in cases that the PMV index equals zero. The PPD index describes the percentage of occupants that are dissatisfied with the given thermal conditions. Statistical data show 5% PPD is the lowest percentage of dissatisfied practically achievable since providing an optimal thermal environment for every single person is not possible. Even though the PMV model works well for adults it requires adjustment for older adults, children and disabled people. Feldmeier [8] demonstrated that using the PMV index is not the best way to calculate thermal comfort. Fanger was aware of the limitations and shortcomings of his model. He advised careful use of the PMV index for values below -2 and above +2. He also warned that the results of the PMV model should be regarded as a first estimation. In practice, while the PMV index has been validated by field studies, these studies pointed out that PMV index should be used with caution when dealing with small groups of people. One of the problems of the PMV index results from the fact that the theory was developed in precisely controlled and monitored environments. Certain values such as clothing insulation that can be easily determined in the context of the experiments are difficult to determine in a real life applications.

International norms and standards provide guidelines regarding the regulation of thermal comfort. Three main guidelines exist, namely, the International Organization for Standardization ISO EN 7730 – 2005a, the American Society of Heating, Refrigerating and Air-Conditioning Engineers ASHRAE Standard 55 – 2004, and the European Committee for Standardization CEN CR 1752. All three provide recommendations based on the PMV index determined by Fanger. Thermal comfort is achieved when $PPD \leq 10\%$ and $-0{,}5 \leq PMV \leq 0{,}5$. It is recommended that no local thermal discomfort factors such as radiant temperature asymmetry, vertical air temperature difference, floor surface temperature, temperature variation with time, etc. are in effect. The standard for thermal comfort is defined by the operative temperature. The operative temperature is defined as a uniform temperature of a radiantly black enclosure in which an occupant would exchange the same amount of heat by radiation plus convection as in the actual non-uniform environment. The operative temperature intervals vary by the

 ISBN: 978-1-61208-308-7

type of indoor location and by the time of year. ASHRAE standards [9] have listings for suggested temperatures and air flow rates in different types of buildings and different environmental circumstances. Moreover, to overcome the limitations of the PMV index, ASHRAE [9] introduced a new adaptive model for measuring thermal comfort. The adaptive hypothesis predicts that contextual factors and past thermal history modify building occupants' thermal expectations and preferences. The ASHRAE-55 2010 Standard has introduced the *prevailing mean outdoor temperature* as input variable for the adaptive model. It is based on the arithmetic average of the mean daily outdoor temperatures (DBT) over no fewer than 7 and no more than 30 sequential days prior to the day in question. This model applies especially to occupant-controlled, natural conditioned spaces, where the outdoor climate can actually affect the indoor conditions and so the comfort zone. Studies by de Dear and Brager [10] showed that occupants in naturally ventilated buildings were tolerant of a wider range of temperatures. ASHRAE Standard 55-2010 states that differences in recent thermal experiences, changes in clothing, availability of control options and shifts in occupant expectations can change people thermal responses. The new model is proposed alongside Fanger's PMV index in every release of ASHRAE Standard 55, however, it has been also ignored by the majority of the HVAC industry.

Further research on thermal comfort considers the heat balance of the human body and calculates sensation and comfort for local body parts [11], [12].

III. PMV AND PPD CALCULATIONS

A MATLAB function was devised to calculate the thermal comfort levels at the interior space of the prototype house. The motivation driving these calculations was to understand the thermal conditions at the house interior on a specific day of the year, based on specific assumptions. The function that calculates thermal comfort contains two parts, namely, the main PMV function named PMV.m., and the ComputeTCL.m function. The PMV.m. function collects the values of the activity, the clothing insulation, the air temperature, the mean radiant temperature, the relative humidity, the air velocity, and the external work. Moreover, PMV.m. has as task to: compute all the necessary values that are needed in order to calculate the PMV and PPD values, call the ComputeTCL.m function to compute the numeric value, and display the PPD on a graph. The ComputeTCL.m function calculates iteratively the clothing temperature value. In the next calculations the values for the metabolism, the insulation induced by the clothes, and the interior air velocity, were provided. The simulation experiments were run for the hottest day of the year, July 15th, in N. Italy assuming that the activity of the person was light and that the person was wearing light clothes such as a pair of shorts and a light shirt. The air velocity was set to the one recommended by the ASHRAE for optimum indoor comfort.

Table II presents the conditions of the experiments I, II, III and IV.

TABLE II. CONDITIONS OF EXPERIMENTS I, II, III, IV

Parameter	I	II	III	IV
Air temperature (C°)	35	24	43	24
Mean radiant temperature	35	30	43	33
Relative humidity (%)	46	55	30	53
Clothing index (Clo)	0.6	0.6	0.6	0.6
Metabolism (*W*)	70	70	70	70
Air velocity (in $m.s^{-1}$)	0.15	0.15	0.15	0.15

A. Experiment I

In the first experiment, the thermal comfort was computed for the hottest day of the year, July 15, at 2:00 PM, in N. Italy assuming that the electrochromic material of the windows was active and the AC system was inactive. Based on the values of Table II, the results were PMV=3 and PPD=100. A plot of the PPD (PMV) appears in Figure 1.

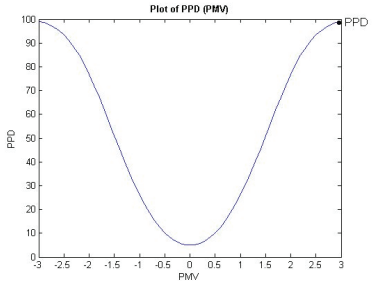


Figure 1. Plot of PPD (PMV) for experiment I.

B. Experiment II

In the second experiment, the thermal comfort was computed for the hottest day of the year, July 15, at 5:00 PM, in N. Italy assuming that the electrochromic material of the windows was active and the AC system was active, too. Based on the values of Table II, the results were PMV=0.17 and PPD=5.6. A plot of the PPD (PMV) appears in Figure 2.

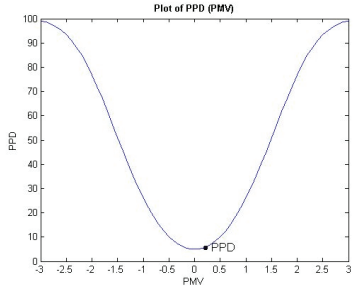


Figure 2. Plot of PPD (PMV) for experiment II

C. Experiment III

In the third experiment, the thermal comfort was computed for the hottest day of the year, July 15, at 2:00 PM, in N. Italy assuming that the electrochromic material of the windows was inactive and the AC system was inactive, too.

Based on the values of Table II, the results were PMV=3 and PPD=100. A plot of the PPD (PMV) appears in Figure 3.

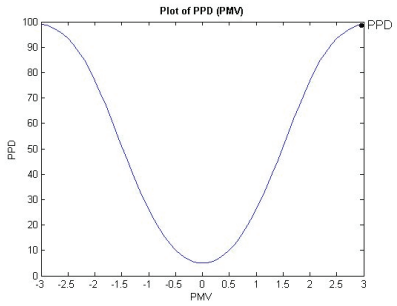


Figure 3. Plot of PPD (PMV) for experiment III.

D. Experiment IV

In the fourth experiment, the thermal comfort was computed for the hottest day of the year, July 15, at 5:00 PM, in N. Italy assuming that the electrochromic material of the windows was inactive and the AC system was active. Based on the values of Table II, the results were PMV=0.47 and PPD=9.6. A plot of the PPD (PMV) appears in Figure 4.

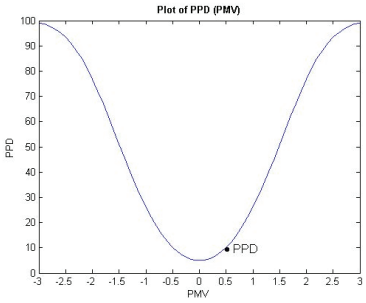


Figure 4. Plot of PPD (PMV) for experiment IV.

The calculations I, III – performed during different time of the day – allow the comparison of performance with and without activating the electrochromic windows. The results indicate an improvement when the electrochromic windows are active, but in both calculations I, III the thermal comfort is far from being achieved. In calculations II and IV, the HVAC system is active. This allows evaluating the impact of a modern HVAC system during the hottest day of the year. In both cases II, IV the PPD value is sufficiently low and statistically 90 % of people will experience thermal comfort.

In the experiments I, II, III and IV, thermal comfort is computed during the most demanding weather conditions. Given that the goal of the prototype house in Trento was not to perform as a passive house, but rather as a low consumption one, even with the parallel operation of an HVAC system, this goal is still attainable. Furthermore, Fanger recognized that the PMV model is not infallible for values superior to +2. In the light of these findings, further study was conducted to allow for a wider spectrum of possible models to offer more accurate account of thermal comfort.

IV. FUZZY LOGIC CONTROLLER

After performing simulations with the MATLAB function, the methods of the calculation were upgraded. More specifically, tools involving fuzzy logic were intergraded to compute the PMV and PPD values with greater flexibility and accuracy. The idea to use a simulation tool based on fuzzy logic originates from a paper by J. van Hoof [13]. Unlike Boolean logic, fuzzy logic allows for additional states than *true* or *false*. The core of the fuzzy logic theory is the membership function. A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1.

The fuzzy logic PMV controller was first discussed in Hamdi et al. [4]. The paper points to several advantages of a fuzzy logic controller compared to other methods. One of the main problems regarding Fanger's PMV model is that his equations are not linear and hence they are unsuitable for a feedback control system [14], [15]. A lot of effort has been made to find a way around the iterative process of calculating the value of the temperature of clothing [16], [17], [18], by simplifying the equations. However, researchers pointed out that such simplifications in the PMV model made it unreliable and subject to errors [15]. Hamdi et al. [4] proposed a way of computing the PMV index accurately without having to use the iterative process. The proposed fuzzy logic controller is based on [4].

The next sections expose the attempt of integrating fuzzy logic in our thermal comfort controller. The main advantages of using fuzzy logic to compute the PMV index of a particular dataset are the use of a non-iterative process that is appropriate for feedback control, the real time simulation capability, and the great accuracy. MATLAB was used again to implement the fuzzy PMV index described in [4]. MATLAB includes a built-in fuzzy logic toolbox that can interact with the SIMULINK environment through a specially designed block, namely, the fuzzy controller block. SIMULINK is built-in the MATLAB environment and can either drive or be scripted from it. It was decided to use the environment for its tight integration with MATLAB, and therefore with the fuzzy toolbox. An additional reason for selecting SIMULINK was the existence of a block library for fuzzy logic. By providing a graphical simulation tool, SIMULINK made the development easier and allowed the simulation of dynamic systems. The final system combines two parts working together to compute the PMV index value, a *personal-dependent model* and an *environmental model*.

A. Personal-Dependent Model

The personal-dependent model takes into consideration only the variables depending on a person inside a room. There are only two inputs for this part of the system, the metabolic activity of the person in question, and his or her clothing index (i.e., clothing insulation). The output of this subsystem is a temperature value contained in a range of temperatures within which stands the operative temperature. The operative temperature is the required temperature to reach a PMV value equal to 0 when the air temperature equals to the mean radiant temperature ($T_{air} = T_{mrt}$) and

RH=50%. The "fuzzy Personal-Dependent Model" block of Figure 5 contains a set of membership functions and their associated rules. Its output is the air temperature range in which the PMV is found to be close to zero. It returns a value contained in a specific range.

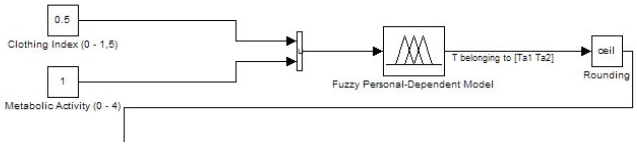


Figure 5. Diagram of the personal-dependent model in SIMULINK

Given that the various temperature intervals are fixed and do not overlap, knowing a specific value provides access to the temperature range in which the value is contained. The purpose of the block immediately following the fuzzy controller block is to round up the output value of the preceding block in order to use this value in the next block (i.e., the switch block) which accepts only integers.

B. *Environmental Model*

While the personal-dependent model is complete, the environmental model is not yet fully operational. Figure 6 presents the block diagram of the environmental model in its current state of development. When completed, this model will be fairly complex. The first block of the environmental model is a "switch block" that allows the system to select between five different temperature ranges, depending on the unique value returned by the first subsystem.

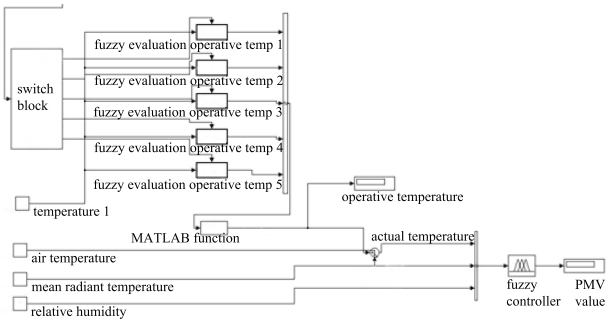


Figure 6. Block diagram depicting the environmental model in its current state of development.

Each block in the middle of the model is a subsystem that contains a fuzzy controller, an input and an output. Figure 7 illustrates a block of this kind.

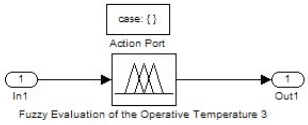


Figure 7. Block diagram depicting a specific subsystem.

Depending on the input value in the "switch block", the corresponding subsystem representing the correct air temperature range is triggered. The input value of each fuzzy controller is the air velocity. In the current state of the system, it is considered constant, but it can be upgraded to produce a signal mimicking the air velocity variations over a period of time. The output value is a temperature displayed in a display block and named "operative temperature". The last block is a MATLAB function (.m function) designed to extract the only positive value contained in the vector created on the right of the five subsystem blocks. In this fashion the system can deal with a number instead of a vector containing five components.

C. *Membership Function of Personal-Dependent Model*

Each fuzzy controller block is embedded in the overall system with a membership function created with the MATLAB fuzzy logic toolbox. The membership functions are identical to the ones described in [4]. Figures 8, 9 and 10 exhibit the membership functions used in the personal-dependent mode of the controller.

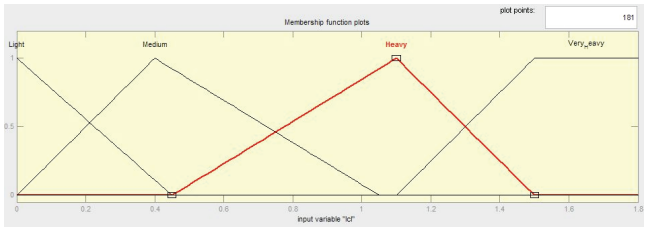


Figure 8. Membership functions of the clothing insulation.

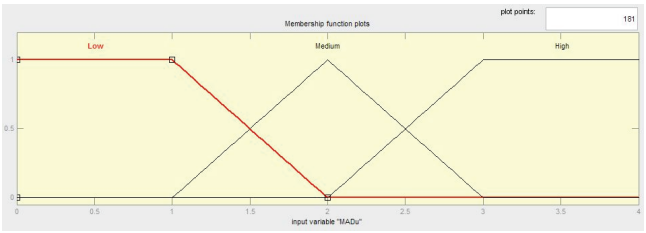


Figure 9. Membership functions of the metabolic activity.

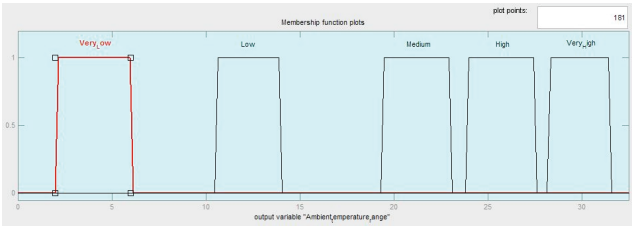


Figure 10. Membership functions of the temperature range.

In the above illustrations of the membership functions of the air temperature range, five temperature intervals are noticeable. The fuzzy controller returns a temperature value as output. Selecting the temperature range within which this value falls is doable since none of the temperature intervals are overlapping. Figure 11 presents the output surface of the

personal-dependent model and visualizes the interaction between the two inputs.

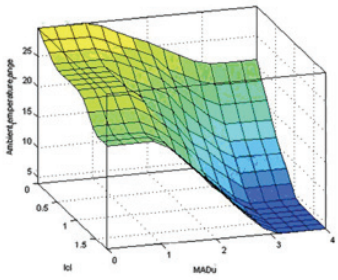


Figure 11. Output surface of the personal-dependent model.

A set of fuzzy rules permits the interaction of the membership functions of the personal-dependent model. Table III presents these rules.

TABLE III. FUZZY RULES OF THE PERSONAL-DEPENDENT MODEL

If |c| is and MADu is then Ambient_temperature_range is
1. If (|c| is Light) and (MADu is Low) then (Ambient_temperature_range is Very_High (1)
2. If (|c| is Light) and (MADu is Medium) then (Ambient_temperature_range is High (1)
3. If (|c| is Light) and (MADu is High) then (Ambient_temperature_range is Medium (1)
4. If (|c| is Medium) and (MADu is Low) then (Ambient_temperature_range is High (1)
5. If (|c| is Medium) and (MADu is Medium) then (Ambient_temperature_range is Medium (1)
6. If (|c| is Medium) and (MADu is High) then (Ambient_temperature_range is Low (1)
7. If (|c| is Heavy) and (MADu is Low) then (Ambient_temperature_range is High (1)
8. If (|c| is Heavy) and (MADu is Medium) then (Ambient_temperature_range is Low (1)
9. If (|c| is Heavy) and (MADu is High) then (Ambient_temperature_range is Very_Low (1)
10. If (|c| is Very_Heavy) and (MADu is Low) then (Ambient_temperature_range is Medium (1)
11. If (|c| is Very_Heavy) and (MADu is Medium) then (Ambient_temperature_range is Low (1)
12. If (|c| is Very_Heavy) and (MADu is High) then (Ambient_temperature_range is Very_Low (1)

Figure 12 presents the screen capturing the rule-viewer of the fuzzy logic MATLAB toolbox. The rule viewer displays in a single screen all the parts of the fuzzy inference process from inputs to outputs. Each row corresponds to a single rule and each column corresponds to an input variable (yellow on the left) or an output variable (blue on the right).

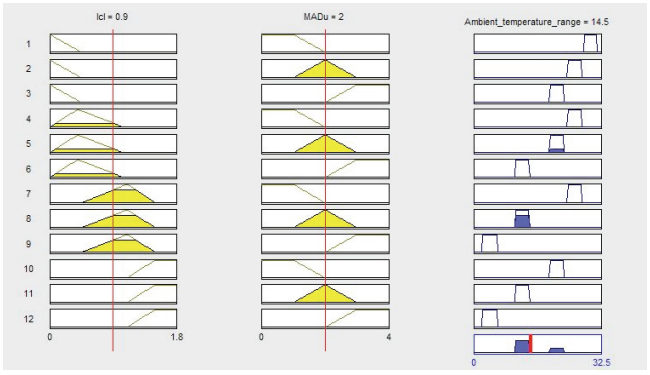


Figure 12. Personal-dependent model rule viewer.

In the case of the environmental model (Figures 13, 14) five fuzzy controllers exist for five temperature ranges. The shapes of the membership functions are identical in each of the cases. The difference resides in the range of the output.

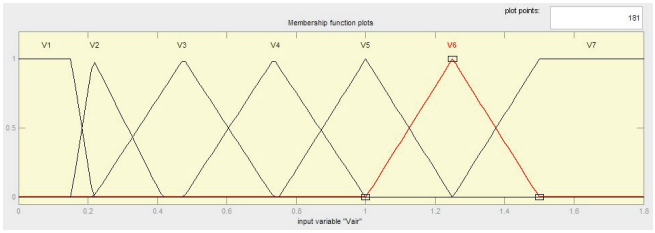


Figure 13. Membership functions of air velocity (environmental model).

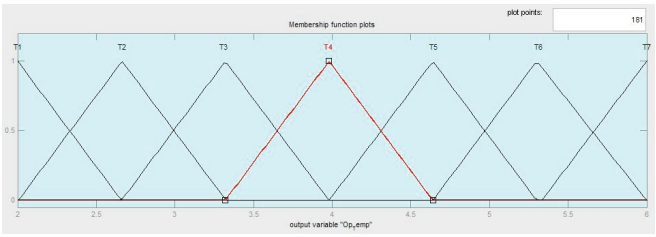


Figure 14. Membership functions of the operative temperature.

The shape of the membership functions is derived from the thermal comfort setting of a user and the interaction of the different parameters. Table IV presents the fuzzy rules of the environmental model that interact with the two previous sets of membership functions.

TABLE IV. FUZZY RULES OF THE ENVIRONMENTAL MODEL

If Vair is then Op_Temp is
1. If (Vair is V1) then (Op_Temp is T1) (1)
2. If (Vair is V2) then (Op_Temp is T2) (1)
3. If (Vair is V3) then (Op_Temp is T3) (1)
4. If (Vair is V4) then (Op_Temp is T4) (1)
5. If (Vair is V5) then (Op_Temp is T5) (1)
6. If (Vair is V6) then (Op_Temp is T6) (1)
7. If (Vair is V7) then (Op_Temp is T7) (1)
8. If (Vair is V8) then (Op_Temp is T8) (1)

Figure 15 presents the screen capturing the rule-viewer of the sets of membership functions.

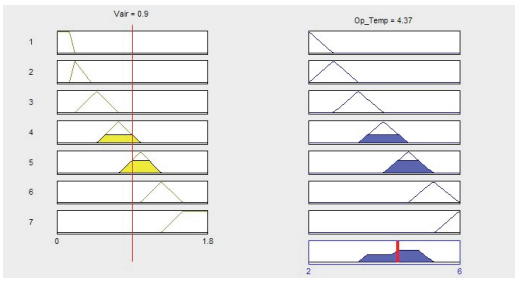


Figure 15. Rule viewer of the environmental model.

D. Implementation Remarks

The implementation of the system encountered two major problems. First problem was that it was necessary to compute the exact operative temperature, even though the temperature range from which it is computed varies each time among a set of five temperature ranges. The adopted solution was to use a "switch block". The "switch block" functions like a switch conditional statement in C#. However, since it accepts only integers as input, this input value is rounded up. Once the value is entered in the "switch block", the "switch block" triggers the sub-system containing the correct fuzzy controller in which the correct membership functions with the correct temperature range are embedded. Since there is no way to know which of the subsystems is going to be triggered, every subsystem returns a value, that can be zero or the operative temperature. A "mux block" allows the system to create a vector containing five components. Since the system can only deal with a single number from this point, a MATLAB function extracts the number corresponding to the operative temperature. In order to accomplish it, the MATLAB function sums all the components of the vector since they are all equaled to zero except from the only one of interest.

The second problem relates to the environmental model. Once the operative temperature is returned by the system, it is compared with the actual air temperature. The effects of the mean radiant temperature and of the relative humidity are considered to adjust the operative temperature – which must be close to the required temperature – in order to reach a PMV value of zero. The problem resides in the range of the membership functions. In order to adjust the temperature the range of the membership function must be regenerated depending on the value of the operative temperature returned by the first part of the environmental model. No solution has been found for this problem.

V. VALLIDATION OF THE SYSTEM

The current system cannot compute a PMV index value in the way it was intended. But, it does compute the operative temperature. The operative temperature is the air temperature for which the PMV index approaches zero when $T_{air} = T_{mrt}$ and $RH = 50\%$. According to Chamra et al. [19], clothing insulation, metabolic activity as well as air velocity are the three fundamental parameters that affect thermal comfort. Hence, PMV is more sensitive to these three parameters than to the mean radiant temperature and the relative humidity. To confirm this statement, a series of calculations of the PMV index value were conducted, with the motivation to prove that the system in its current state, while not fully operational is still competent. It simulates PMV and it can supply a temperature close to thermal comfort. The simulations of the Tables V, VI, VII and VIII, follow four assumptions:

a) Relative humidity falls rarely below 20% or above 80%.
b) Optimum air velocity is 0,15 $m.s^{-1}$.
c) Comfort zone is $-0,5 \leq \text{PMV} \leq 0,5$ and
d) Metabolic rate is 1 *met*. Such value is consistent with a sedentary light activity at a house interior.

The Tables V, VI, VII, and VIII contain the results of the simulations that were performed in order to:

(i) Confirm that the operative temperature yields PMV close to zero if $T_{air} = T_{mrt}$ and $RH = 50\%$ with variable clothing insulation.

(ii) Determine the impact of the mean radiant temperature and the relative humidity on the PMV value for $T_{air} = T_{operative}$.

TABLE V. RESULTS OF 8 SIMULATION EXPERIMENTS WITH CLOTHING INSULATION = 1 CLO

Parameters	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6	Exp 7	Exp 8
Clothing (clo)	1	1	1	1	1	1	1	1
Operative temp. (°C)	24,2	24,2	24,2	24,2	24,2	24,2	24,2	24,2
Mean radiant temp. (°C)	24,2	23,2	22,2	21,2	20,2	19,2	18,2	17,2
Activity (met)	1	1	1	1	1	1	1	1
Air speed (m/s)	0,15	0,15	0,15	0,15	0,15	0,15	0,15	0,15
Relative humidity (%)	50	50	50	50	50	50	50	50
PMV	0,2	0	-0,1	-0,2	-0,3	-0,4	-0,5	-0,6
PPD	5,8	5	5,2	5,8	6,9	8,3	10,2	12,5

TABLE VI. RESULTS OF 8 SIMULATION EXPERIMENTS WITH CLOTHING INSULATION = 1,5 CLO

Parameters	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6	Exp 7	Exp 8
Clothing (clo)	1,5	1,5	1,5	1,5	1,5	1,5	1,5	1,5
Operative temp. (°C)	19,7	19,7	19,7	19,7	19,7	19,7	19,7	19,7
Mean radiant temp. (°C)	19,7	18,7	20,7	21,7	22,7	17,7	16,7	26,7
Activity (met)	1	1	1	1	1	1	1	1
Air speed (m/s)	0,15	0,15	0,15	0,15	0,15	0,15	0,15	0,15
Relative humidity (%)	50	50	50	50	50	50	50	50
PMV	-0,3	-0,3	-0,2	-0,1	0	-0,4	-0,5	0,4
PPD	6,9	6,9	5,8	5,2	5	8,3	10,2	8,3

In all experiments, with the exception of the one presented in the next Table VII, the operative temperature is sufficient to provide thermal comfort when the air temperature equals to the mean radiant temperature ($T_{air} = T_{mrt}$) and $RH = 50\%$. In the experiments presented in Table VII, the clothing insulation is $I_{cl} = 0,5$ *clo* and thermal comfort is not achieved because the PMV is out of the recommended range. However, it should be kept in mind that still only 22% of people would experience discomfort.

Regarding the influence of the mean radiant temperature, in the worst case, there is a difference of 3 C° between the mean radiant temperature for which the PMV equals zero and the mean radiant temperature for which PMV $\geq |0,5|$. This case appears in Table VII in the comparison of the

 ISBN: 978-1-61208-308-7

mean radiant temperature values of Experiment 3 and Experiment 6. This provides a margin of error that makes possible using the system in its present stage of development. According to the simulation data recorded with *Design Builder* software, the mean radiant temperature is always close to the air temperature when the electrochromic windows are activated.

TABLE VII. RESULTS OF 8 SIMULATION EXPERIMENTS WITH CLOTHING INSULATION = 0,5 CLO

Parameters	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6
Clothing (clo)	0,5	0,5	0,5	0,5	0,5	0,5
Operative temp. (°C)	24,2	24,2	24,2	24,2	24,2	24,2
Mean radiant temp. (°C)	24,2	25,2	26,2	27,2	28,2	29,2
Activity (met)	1	1	1	1	1	1
Air speed (m/s)	0,15	0,15	0,15	0,15	0,15	0,15
Relative humidity (%)	50	50	50	50	50	50
PMV	-0,9	-0,7	-0,5	-0,4	-0,2	0
PPD	22,1	15,3	10,2	8,3	5,8	5

TABLE VIII. RESULTS OF 8 SIMULATION EXPERIMENTS WITH CLOTHING INSULATION = 1 CLO AND HIGHER AIR SPEED

Parameters	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6	Exp 7	Exp 8
Clothing (clo)	1	1	1	1	1	1	1	1
Operative temp. (°C)	24,6	24,6	24,6	24,6	24,6	24,6	24,6	24,6
Mean radiant temp. (°C)	24,6	27,6	23,6	22,6	21,6	18,6	22,6	22,6
Activity (met)	1	1	1	1	1	1	1	1
Air speed (m/s)	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2
Relative humidity (%)	50	50	50	50	50	50	0	100
PMV	0,2	0,6	0,1	0	-0,1	-0,5	-0,4	0,4
PPD	5,8	12,5	5,2	5	5,2	10,2	8,3	8,3

Regarding the impact of the relative humidity on the PMV value when $T_{air} = T_{mrt}$, in all cases except one, the PMV stays within the boundaries of thermal comfort PMV ≤ |0,5|. These cases are depicted in Experiment 1 of each of the above Tables, where relative humidity RH = 50 %. The exception appears in the Experiment 1 of Table VII, where PMV = -0,9. These results reveal that although relative humidity does affect PMV, its effect is not critical.

VI. CONCLUSION AND FUTURE WORK

The objective of this study is to produce a system for personalizing thermal comfort at the interior of a prototype house in Trento, N. Italy. A MATLAB function was developed to compute the PMV and PPD values from a given dataset. A first set of calculations during the hottest day of the year proved that the house would provide uncomfortable interior conditions for most people whether or not its electrochromic windows are active. The same calculations also proved that the HVAC system would contribute to the achievement of thermal comfort in the house interior. In order to enhance the accuracy of the thermal comfort calculation and to offer dynamic features, fuzzy logic was used. Fuzzy logic allows computing the PMV in real time thus enhancing the attractiveness of the system for real life application. While the overall system cannot compute a PMV index value as it was intended at the beginning, the development of the thermal comfort controller is still in progress, and the system – given the proper data – produces a correct operational value of thermal comfort for the interior of the prototype. This temperature can be used to adapt the thermal comfort with accuracy.

SIMULINK was used for the graphical interface of the system, to make the design and the implementation of the system less complex. The actual code was also generated by SIMULINK providing the opportunity to further develop the proposed experimental apparatus in a real life system. To that effect, a built-in module for SIMULINK called Real-Time Workshop generated the C code equivalent to the system built with blocks in the graphical interface.

While this paper addressed the development of a computational apparatus calculating the thermal comfort of a person, it does not touch upon the implementation of an actual house system. The development of such a system that computes thermal comfort is in progress. The next steps of this research will be to complete the environmental subsystem and adjust the membership functions in order to obtain better accuracy. Once the environmental subsystem is completed, some further improvements could be made in order to fully take advantage of the possibilities offered by the MATLAB and the SIMULINK environment. Possible enhancements may include adding a feedback loop that would make the system adaptive. Adding a neural network would make the system capable of learning the preferences of its users [20], thus transforming the fuzzy PMV system into a neuro-fuzzy PMV system. Furthermore, it is possible to use signal generators instead of using constant values as inputs. This would take advantage of the capability of SIMULINK for conducting dynamic simulations, thus providing insights on how the PMV and the operative temperature vary during a certain period of time. Experimenting with different membership functions could be another option. Triangular functions were used in the current implementation. The use of Gaussian or polynomial based membership functions could contribute to better accuracy.

Finally, since the calculation of the PMV index value is based on statistics there are no guarantees that the thermal conditions decided by the computer will satisfy the inhabitants. Nonetheless, the proposed model can predict what the end user will be expecting. To radically improve the system, this must be paired with a learning capacity. Then given time and feedback, the system will learn from the end-user experience, and refine the PMV model to accommodate the needs of the inhabitants. In order to learn and adapt based on the requirements of the users, the computer will have to

 ISBN: 978-1-61208-308-7

collect feedback data. And the error margin will decrease, as the user's inputs will be stored into the system.

ACKNOWLEDGMENT

This research was conducted within the Green Home Alliance between the Mobile Experience Lab at the Massachusetts Institute of Technology and the Fondazione Bruno Kessler, in Trento, N. Italy.

REFERENCES

[1] P. O. Fanger, Thermal Comfort, Analysis and Applications in Environmental Engineering, McGraw-Hill, 1972.

[2] S. Lynch, Dynamical Systems with Applications using MATLAB, Birkhäuser, 2004.

[3] MATLAB SIMULINK: Simulation and Model Based Design, http://www.mathworks.com

[4] M. Hamdi, G. Lachiver, and F. Michaud, "A new predictive thermal sensation index of human response", Energy and Buildings, Volume 29, Issue 2, 1999, pp. 167-178.

[5] S. D. Kotsopoulos, F. Casalegno, M. Ono, and W. Graybill, "Managing Variable Transmittance Windowpanes with Model-Based Autonomous Control", Journal of Civil Engineering and Architecture, Issue 5, Vol. 7, (serial No 66), May 2013, pp. 507-523.

[6] S. D. Kotsopoulos, F. Casalegno, M. Ono, and W. Graybill W., 2012, "Window Panes Become Smart: How responsive materials and intelligent control will revolutionize the architecture of buildings", Proceedings of the First International Conference on Smart Systems, Devices and Technologies, Stuttgart, Germany, pp. 112-118.

[7] H. O. Nilsson "Comfort Climate Evaluation with Thermal Manikin Methods and Computer Simulation Models", Indoor Air, Volume 13 Issue 1, March 2003, pp. 28-37.

[8] M. C. Feldmeier, 2009, "Personalized Building Comfort Control", PhD thesis, Department of Architecture and Planning, Massachusetts Institute of Technology, 2009.

[9] ASHRAE handbook—fundamentals, Chap. 8, Physiological Principles, Comfort and Health, 1989.

[10] R. de Dear and G. S. Brager, "Developing an adaptive model of thermal comfort and preference", ASHRAE Transactions, 104(1a), 1998, pp. 145-167.

[11] H. Zhang, E. Arens, C. Huizenga, and H. Taeyoung, "Thermal sensation and comfort models for non-uniform and transient environments: Part I: local sensation of individual body parts", Indoor Environmental Quality, UC Berkeley, 2009.

[12] H. Zhang, E. Arens, C. Huizenga, and H. Taeyoung, "Thermal sensation and comfort models for non-uniform and transient environments: Part II: local comfort of individual body parts", Indoor Environmental Quality, UC Berkeley 2009.

[13] J. van Hoof, "Forty years of Fanger's model of thermal comfort: comfort for all?", Indoor Air, Volume 18, Issue 3, June 2008, pp. 182-201.

[14] D. Int-Hout, 1990, "Thermal comfort calculations / A computer model", ASHRAE Transactions 96 (1), 1990, pp. 840–844.

[15] C. C. Federspiel, "User-adaptable and minimum-power thermal comfort control", PhD thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, 1992.

[16] A. Auliciems, "Thermobile controls for human comfort", The Heating & Ventilating Engineer, April/May 1984, pp. 31–33.

[17] D. J. Coome, G. Gan, and H. B. Awbi, "Evaluation of thermal comfort and indoor air quality", Proceedings of CIB'92 World Building Congress, Montreal, 1992, pp. 404–406.

[18] C. H. Culp, M. L. Rhodes, B. C. Krafthefer, and M. A. Listvan, "Silicon infrared sensors for thermal comfort and control", ASHRAE Journal, April 1993, pp. 38–42.

[19] L. M. Chamra,W. G. Steele, and K. Huynh, "The uncertainty associated with thermal comfort", ASHRAE Transactions, 109, 2003, pp. 356–365.

[20] L. X. Wang and J. M. Mendel, "Back-propagation fuzzy system as nonlinear dynamic system identification", in Proceedings of IEEE International Conference on Fuzzy Systems, 1992, pp. 1409–1418.

 ISBN: 978-1-61208-308-7

The Impact of Control Setpoints on Building Energy Use

Stephen Treado, Xing Liu
Department of Architectural Engineering
Pennsylvania State University
University Park, PA, USA
Emails: {streado, xul121}@psu.edu

***Abstract*—This paper examines the impact of building Heating, Ventilation and Air Conditioning (HVAC) control system setpoints such as temperature and flow rate on total building energy requirements, for a typical system design and operation. Through the analysis focused on a summer and winter operating condition, the range of energy usage and the potential for minimizing building energy requirements by dynamically adjusting setpoints are presented in this paper.**

Keywords-buildings; cooling; control systems; energy; heating; HVAC; optimization

I. INTRODUCTION

The increasing demand of air-conditioning and the energy crisis during the last decades have led to a surge of attention and there is no doubt that the improvement of the Heating, Ventilating and Air Conditioning (HVAC) control system is one of the effective solutions to realize sizable energy-saving for the building sector. The aim of HVAC control is to provide a comfortable, safe, healthy and productive environment for occupants using the least energy. Significant energy saving potential exists for building systems during operation with the help of current technology such as intelligent, adaptive or model predictive control. The development of this kind of technology has led to the possibility of the improvement of building operational performance. However, it is difficult to evaluate the potential or effectiveness of the new control strategies without first gaining a better understanding of the range of operating conditions possible for any particular building/HVAC system combination. That is, the amount of energy savings is a function of both the actions of the new control strategy and the fundamental capabilities of the HVAC system. In its most basic form, a building control system can do no more than monitor sensors, apply logic and manipulate actuators. Thus, the main objective of the work described in this paper is to clearly identify and define the space within which the building/HVAC combination is capable of operating in order to enable the determination of both energy saving potential and optimal setpoints and control logic. While this is not specifically an optimization effort, i.e. we are not seeking a single optimal solution since it is understood that setpoints and control logic may need to be adjusted on a dynamic basis, the primary metric utilized, namely total building energy usage, can be considered as an objective function.

The content is organized as follows. Section II reviews the recent studies. Section III presents the models adopted and simulation work. Section IV gives the results and Section V presents the conclusions.

II. LITERATURE RIEVEW

Simulation is taken as one of the oldest but very effective tools to engineers in every discipline. Building simulation began in the 1960s and became the hot topic of the 1970s within the energy research community. For nowadays, computer simulation is not only used for the building design stage like sizing and configuration design, but also adopted for system performance analysis more and more widely. Building simulation can be applied to reveal the inter-actions between the building itself and its occupants, HVAC systems, and the outdoor climate. A large amount of work has been done to show how important building simulation is in the study of energy performance and the design and operation of energy-efficient buildings [2]. For examples, Li et al. [8] and Pan et al. [12] analyzed and displayed the building energy break-down with calibrated models in 2007 and 2009, respectively; however, more effort is needed to understand how to obtain optimum operating parameters, particularly for building control systems. Simulation does provide a good opportunity to evaluate the dynamic and energy performance of HVAC system control strategy in a convenient and low cost way. The control strategy can also be pre-tuned before being utilized in the real system with the help of simulation. Recent research also showed performing building simulation analysis enabled diagnosis of malfunctioning or incorrectly commissioned equipment within the building and thus also assisted with future commissioning and tuning of the building performance [11].

Future development and application of information technology in the building industry will lead to a completely new building design philosophy and methodology [7]. In 2003, Mathews and Botha [9] conducted simulation with three cases and proved that simulation does indeed have the ability to improve the thermal and energy management of building HVAC systems. A lot of work has been done in the

 ISBN: 978-1-61208-308-7

TABLE I. REFERENCE CHARACTERISTICS OF EQUIPMENT

Components	Selected parameters values			
Chiller	25000	Capacity (W)	2.75	COP
	44	T_lcw (°F)	85	T_ecf (°F)
	111.7	V_chw (gpm)	128.5	V_cdw (gpm)
Natural Gas Boiler	0.8	Boiler Efficiency	950 (Btu/lb)	Heat Value
Variable Volume Fan	4500 (gpm)	Rated Flow rate	1837	Rated Power (W)
	600 (Pa)	Pressure Rise	0.7	Fan Efficiency
Variable Speed Pump	67.02 (gpm)	Rated Flow rate	500	Rated Power (W)
	50	Pump head (ft)	0.66	Pump Efficiency

*T-Temperature, V-Flow Rate, lcw-leaving chilled water, ecf-entering condenser fluid, chw-chilled water, cdw-condenser water

TABLE II. DEFAULT PARAMETER VALUE FOR SIMULATION

Variable	Value
Zone Area	S=750 ft^2
Overall Envelope Heat Transfer Rate	UA = 0.3 Btu/h-ft^2-°F
Ambient Temperature	T_a = 90 °F (summer condition)
	T_a = 30 °F (winter condition)
Ambient Pressure	P = 101 atm
Zone Air Temperature	T_z = 75 °F (summer condition)
	T_z = 72 °F (winter condition)
Outdoor Air fraction	F_o = 70%
Solar Heat Gain	q_S =1.5 w/ft^2 (summer condition)
	q_S =0.8 w/ft^2 (winter condition)
Lighting Heat Gain	q_l = 1.0 w/ft^2
Equipment Heat Gain	q_e = 1.0 w/ft^2
Occupants Heat Gain	q_o = 1.0 w/ft^2
Ventilation Air Flow rate	M_v = 1.5 cfm/ft^2
Infiltration Air Flow Rate	M_i = 0.1 cfm/ft^2
Heat Exchanger Effectiveness	U_1 = 75%
Energy Recovery Effectiveness	U_2 = 70%

field of building energy consumption simulation but more work remains to be done. Traditionally, less attention has been put on buildings operation compared with the design of a system and its construction/installation. What's more, the simulation software has been evolving steadily over recent years. HVAC component and subsystem models are now generally well understood and have been the subject of a number of researches [4]. Simulation has been extended to the use to the building operation process, although it has been traditionally regarded as a design tool.

III. SIMULATIONS

The simulations that were conducted consisted primarily of quasi steady state determinations of hourly incremental and total building energy requirements for a range of setpoint combinations and exposed to a summer (cooling) or winter (heating) condition. In essence, a grid was established which represented a collection of setpoints, and annual building energy performance was determined for each grid point. The setpoints were constrained to maintain proper equipment operating conditions (e.g. temperature, mass flow). The primary objective of the simulations was to quantify the range of possible operating points and the maximum potential savings, assuming that the control logic could direct the HVAC system to the optimal operating conditions. Equipment performance was modeled as described below.

Total building energy was determined utilizing performance characteristics of the each component, the chiller, the cooling tower and chiller water pump and the supply air fan plus the energy input value related to lighting and other electrical equipment. The evaluation metric:

$$E_{total} = E_{lighting} + E_{equipment} + E_{chiller} + E_{pump} + E_{fan} \quad (1)$$

where:

E_{Total} = total energy power density

$E_{Lighting}$ = lighting power density input

$E_{Equipment}$ = Equipment power density input

$E_{Chiller}$ = chiller power density input

E_{Pump} = pump power density input

E_{Fan} = fan power density input

The first two terms are specified as follows, according to ASHRAE Standard 90.1 IP [1]:

$E_{Lighting}$ =1.0 w/ft^2

$E_{Equipment}$ =1.5 w/ft^2

The system schematic is presented in Figure 1. As the diagram shows, one zone of a multiple zone Variable Air Volume (VAV) system with energy recovery ventilator was studied for this simulation analysis. For HVAC component energy consumption analysis, polynomial fits were used with representative coefficients, with the important variables being chilled water supply temperature, coil loads, chilled water flow rate, outdoor air fraction, supply airflow rate, supply air temperature and room temperature [6]. These component mathematical equation models are commonly used in similar applications. For the simulation software, Engineering Equation Solver (EES) [5] was selected because of its built–in high-accuracy thermodynamic and heat transfer parameters and capability for solving design problems in which the effects of one or more parameters must be determined. Previous research work shows that the simplicity of the models and the use of an equation solver to run the simulation ensure good robustness and full transparency [3]. Table I summarizes the model parameters.

To minimize the effect from the building itself on the simulation results, the zone is simplified as much as possible. The case that is used in this simulation is assumed to be an office zone has a dimension of 25ft ×30ft with a 9ft high ceiling. An overall envelop thermal transfer rate is

 ISBN: 978-1-61208-308-7

given. The U value is assumed to be 0.3 Btu/h-ft^2-°F. The infiltration rate through the exterior walls is set at 0.1cfm/ft^2, which is based on information from [10]. This infiltration occurs 24 hours a day. The ventilation rate is assumed to be 1.5 cfm/ ft^2. For the lighting, equipment and occupants heat gain are all assumed equal to 1w/ft^2. Also, the effectiveness of the energy wheel is assumed to be constant throughout the year while it is not true in real word. It should change as the outdoor temperature and humidity change throughout the year. For this case, the effectiveness is set at 70% and the effectiveness for the heat exchanger is assumed to be 75%.

Two representative outdoor conditions were analyzed, namely 1) summer condition, and 2) winter condition. And the latent load, which is produced when moisture in the air goes from a vapor to a liquid state, is not calculated in this paper but will be discussed in the future work. In order to evaluate the objective function as defined, it is necessary to specify some parameters first (Table II).

$$Q_z = q_s + q_i + q_t + q_o + q_e + q_l \quad (2)$$

where:
q_s = solar load
q_i = infiltration air load
q_t = envelope thermal load
q_o = occupants load
q_e = equipment load
q_l = lighting load

As shown above, the zone load is made up of solar load, lighting load, equipment load, occupants load, infiltration air load and envelope thermal load (heat gains to zone were assumed as positive). The zone heating and cooling loads are met by supplying conditioned air to the zone such that the product of the mass flow rate of the supply air, the specific heat of air and the temperature change of the air from supply (T_s) to return (T_r) are equal to the zone thermal load:

$$q_i = m_i \cdot cp_{air} \cdot (T_z - T_a) \quad (3)$$

$$q_t = UA \cdot (T_z - T_a) \quad (4)$$

Since the heat gain from lighting, equipment occupants and solar was already set up, the load values of infiltration and envelope thermal conduct can be determined from the thermodynamic relationships as described above, the zone load can be figured out for the energy consumption simulation.

For the summer condition simulation, five parameters: condenser entering temperature, chilled water supply temperature, chilled water mass flow rate, supply air temperature and flow rate are set as variables. Ten different values are selected for each parameter so there are 50 different scenarios in total. As only hot water supply temperature and mass flow rate, supply air temperature and flow rate were changed in the winter condition, 40 group of total power density resulted from the simulation. The component energy consumption was simulated with polynomials, as described below:

$$E_{chiller} = \frac{Q_{avail} \cdot ChillerEIRFTemp \cdot ChillerEIRFPLR}{COP_{ref}} \quad (5)$$

$$E_{pump} = v_{water} \cdot \frac{Pump_{Head}}{Total_{Efficiency}} \quad (6)$$

$$E_{fan} = f_{pl} \cdot m_{design} \cdot \frac{P_{rise}}{e_{tot} \cdot r_{air}} \quad (7)$$

where:
$Q_{avail} = Q_{ref} \times ChillerCapFTemp$
V_{water} = mass flow rate of chilled/hot water
f_{pl} = air part load factor
m_{design} = fan design flow rate
P_{rise} = fan pressure rise
E_{tot} = fan total efficiency
ρ_{air} = density of air

In the heating situation, the fuel input was calculated with this equation [13]:

$$F_{boiler} = m_{hw} \cdot cp_{water} \cdot \left[\frac{T_{hws} - T_{hwr}}{BE \cdot VHI}\right] \cdot 3600 \quad (8)$$

where:
BE = boiler efficiency
VHI = fuel heat value
m_{hw} = hot water mass flow rate
cp_{water} = specific heat capacity of water
T_{hws} = hot water supply temper
T_{hwr}= hot water return temperature

TABLE III. CASE DESCRIPTION FOR THE TWO CONDITIONS

Cases(summer)	Simulation Description	Results Range
1 (group 1-10)	Increase condenser entering temperature (50-68 °F)	4.87-5.21 w/ ft^2
2 (group11-20)	Increase chilled water supply flow rate (0.4-0.7 lbm/s)	4.70-5.40 w/ ft^2
3 (group 21-30)	Increase chilled water supply temperature (41-59 °F)	5.14-5.12 w/ ft^2
4 (group 31-40)	Increase supply air flow rate (0.4-0.7 lbm/s)	4.57-5.27 w/ ft^2
5 (group 41-50)	Increase supply air temperature (59-68 °F)	5.24-4.75 w/ ft^2

Cases(winter)	Simulation Description	Results Range
1 (group 1-10)	Increase hot water supply temperature (176-194 °F)	5.95-5.61 w/ ft^2
2 (group11-20)	Increase hot water supply flow rate (0.4-0.7 lbm/s)	5.60-5.36 w/ ft^2
3 (group 21-30)	Increase supply air flow rate (0.6-0.8 lbm/s)	5.36-5.16 w/ ft^2
4 (group 31-40)	Increase supply air temperature (85-92 °F)	5.12-4.88 w/ ft^2

 ISBN: 978-1-61208-308-7

IV. RESULTS

Figure 2 illustrates the power density for five different cases from largest to the smallest in the summer condition. The different colors indicate the breakdown of the electricity usage. Lighting and equipment represent fixed loads, while chiller, pump and fan energy, respectively, vary in response to the each specific combination of setpoints. Variation in total building energy for the summer condition is 18%. This indicates that use of the best setpoint combination could achieve an 18% reduction in total building energy compared to the worst setpoint combination. As we can see, HVAC system (including chiller, cooling tower pump, chiller water pump and supply air fan) is the biggest electric consumer in the building, which accounts for around 45% of total energy consumption, while lighting and equipment account for around 22% and 33% of the total electricity consumption, respectively. According to Table III, the maximum power density can reach 5.40w/ft^2 when the chilled water flow rate at the biggest value and a small supply air flow rate can decrease the energy consumption to 4.57w/ft^2. These calculations could be repeated at any desired interval to enable the continuous reassessment and adjustment of setpoints.

Figure 3 illustrates the power density for four different cases from the largest value to smallest value in winter condition. In this case, the tradeoff is between boiler fuel inputs, pump and fan power. As the natural gas boiler replaced the electrical chiller for conditioning the zone temperature, the electricity usage is decreased, because cooling tower pump is not needed, so the pump energy percentage is also reduced. As a result, the HVAC system (including pump and fan) only accounts about 30% of total electricity consumption.

The best and worst scenarios happened when the hot water flow rate is the highest and when the supply air flow rate is lowest respectively, which was similar to the results for the summer condition. The largest power density is 5.95w/ft^2 and the smallest value is 4.88 w/ft^2 based on Table III. The maximum potential savings due to setpoint manipulation for the winter condition was 22%. As before, this process can be repeated at any desired time interval to allow continuous dynamic adjustment of setpoints to achieve maximum energy efficiency.

The energy performance of this particular building/HVAC system combination was evaluated for typical summer (cooling) and winter (heating) scenarios in order to illustrate the methodology and the energy saving potential of dynamic setpoint manipulation. While the magnitude of the potential energy savings would be expected to vary for different buildings and locations, the methodology would still be applicable and useful provided the proper information was available to accurately model the HVAC system and its components. The methodology could also be used to evaluate the effectiveness of advanced control strategies by comparing the energy savings predicted or realized by those methods to the maximum potential savings identified using the approach described here.

V. CONCLUSION

A methodology was developed and demonstrated for determining the impact of HVAC control system setpoints on the total building energy requirements for different building operation situations in the cooling and heating seasons in order to quantify the maximum potential energy savings due to dynamic setpoint adjustment. According to the simulation result, the energy saving potential through possible optimum control is substantial and more noticeable in winter season. The potential saving can be as high as 18% and 22% for cooling and heating, respectively, when comparing the best performance with the worst one. Different control system setpoints provide different degree of energy savings. Minimizing the supply air flow rate is shown to be the most effective measure to save electricity usage in both cooling and heating season, while a large chilled/hot water flow rate will consume the most power. The results suggest that control strategies that are capable of dynamically adjusting setpoints in response to environmental and occupant conditions can potentially save a substantial amount of energy as compared to fixed setpoints.

REFERENCES

[1] ASHRAE, 2010. ASHRAE Standard 90.1-2010, Energy standard for buildings except low-rise residential buildings, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc.

[2] S. Bertagnolio, P. Andre, and V. Lemort, “Simulation of a building and its HVAC system with an equation solver: Application to audit,” Building Simulation, Volume 3, Issue 2, pp. 139-152, June 2010.

[3] J. Clarke, J. Cockroft, S. Conner, J.Hand, N. Kelly, R. Moore, T. O’Brien and P. Strachan, “Simulation-assisted control in building energy management systems,” Energy and Buildings, 34(9), pp. 933-940, 2002.

[4] R. C. Clark, “HVACSIM+ Building Systems and Equipment Simulation Program Reference Manual,” Published by the U.S. Department of Commerce, National Bureau of Standards, National Engineering laboratory, Center for Building Technology, Building Equipment Division, Gaithersburg, MD 20899

[5] Engineering Equation Solver, User Manual, F-Chart Software, 1992.

[6] Energy Plus, Engineering Reference Manual, U.S. Dept. of Energy, 2010.

[7] T. Z. Hong, S. K. Chou and T. Y. Bong, “Building simulation: an overview of developments and information sources,” Building and Environment 35 (2000), 2000, pp. 347-361.

[8] Y. Li, Y. Pan and C. Chen, “Study on energy saving retrofitting strategies for existing public building in Shanghai,” Proceedings of Energy Sustainability2009, San Francisco, California, USA, 2009. pp. 64

[9] E. H. Mathews and C. P. Botha, "Improved thermal building management with the aid of integrated dynamic HVAC simulation," Building and Environment 38(12), 2003, pp. 1423-1429.

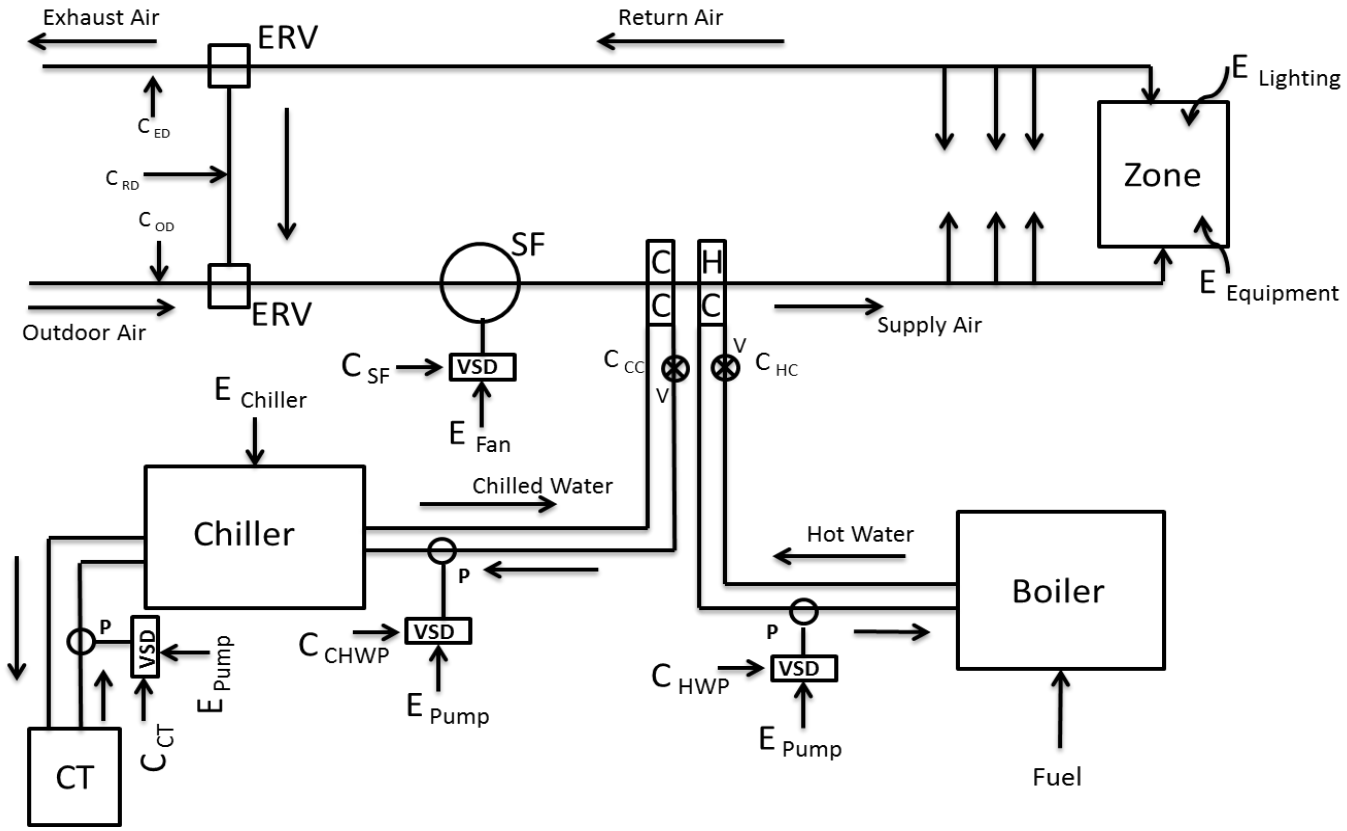
 ISBN: 978-1-61208-308-7

[10] F.C. McQuiston and J.D. Spitler, Cooling and Heating Load Calculation Manual, 2nd ed., American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., Atlanta, Georgia, 1992.

[11] G. Osborne, "The contribution of simulation to the building tuning process for 2 Victoria Avenue," Proceedings of Building Simulation, Sydney, Australia, November 14-16, 2011.

[12] Y. Pan, Z. Huang and G. Wu, "Calibrated building energy simulation and its application in a high-rise commercial building in Shanghai," Energy and Buildings 39 (2007), 2007, pp. 651-657.

[13] A. Wienese, "Boiler, Boiler Fuel and Boiler Efficiency," Proceedings of South African Technology Association, 2001, pp. 75.



Nomenclature	
ERV	Energy Recovery Ventilator
SF	Supply Fan
CC	Cooling Coil
HC	Heating Coil
VSD	Varied Speed Driver
P	Pump
V	Valve

Control Point Lists	
C_{ED}	Exhaust Air Damper
C_{RD}	Return Air Damper
C_{OA}	Outdoor Air Damper
C_{SF}	Supply Fan Driver
C_{CC}	Cooling Coil Valve
C_{HC}	Heating Coil Valve
C_{CT}	Cooling Tower Pump
C_{CHWP}	Chilled Water Pump
C_{HWP}	Hot Water Pump

Figure 1. System Schematic

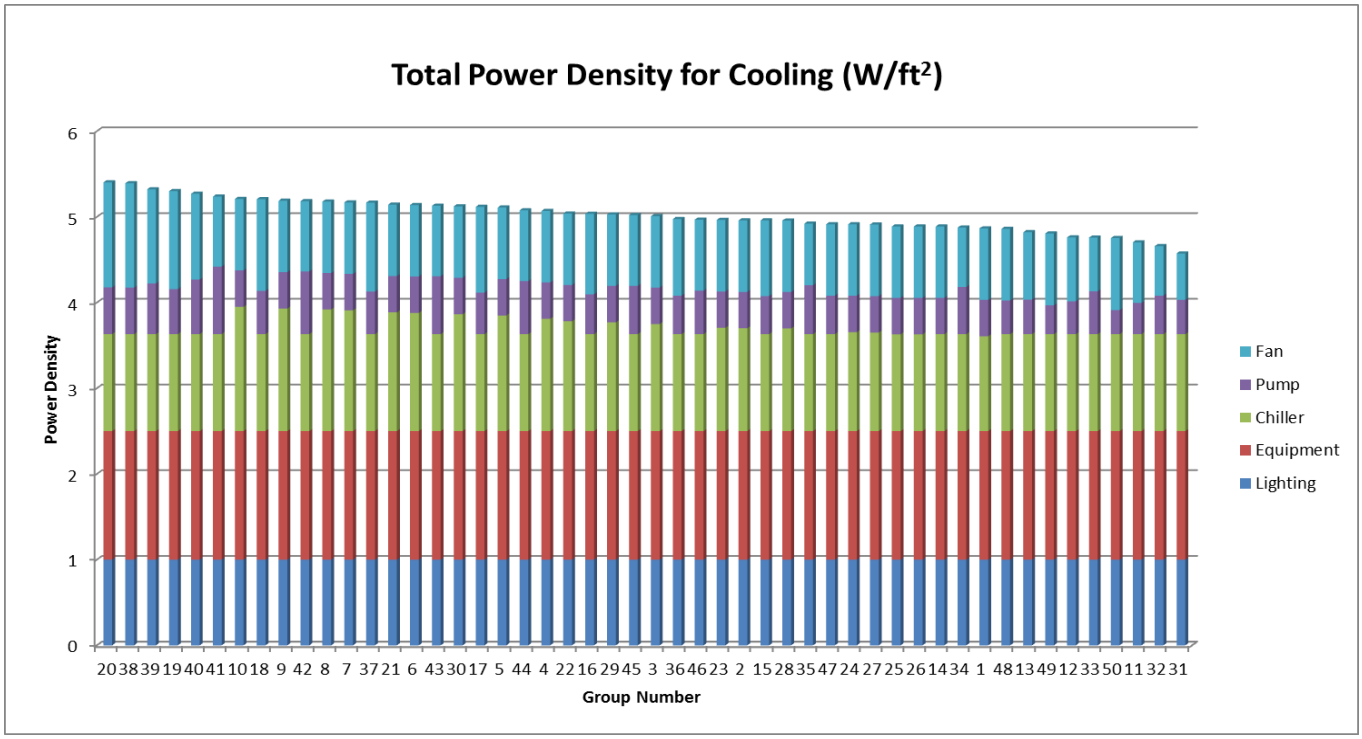


Figure 2. Total Power Density for Summer Condition

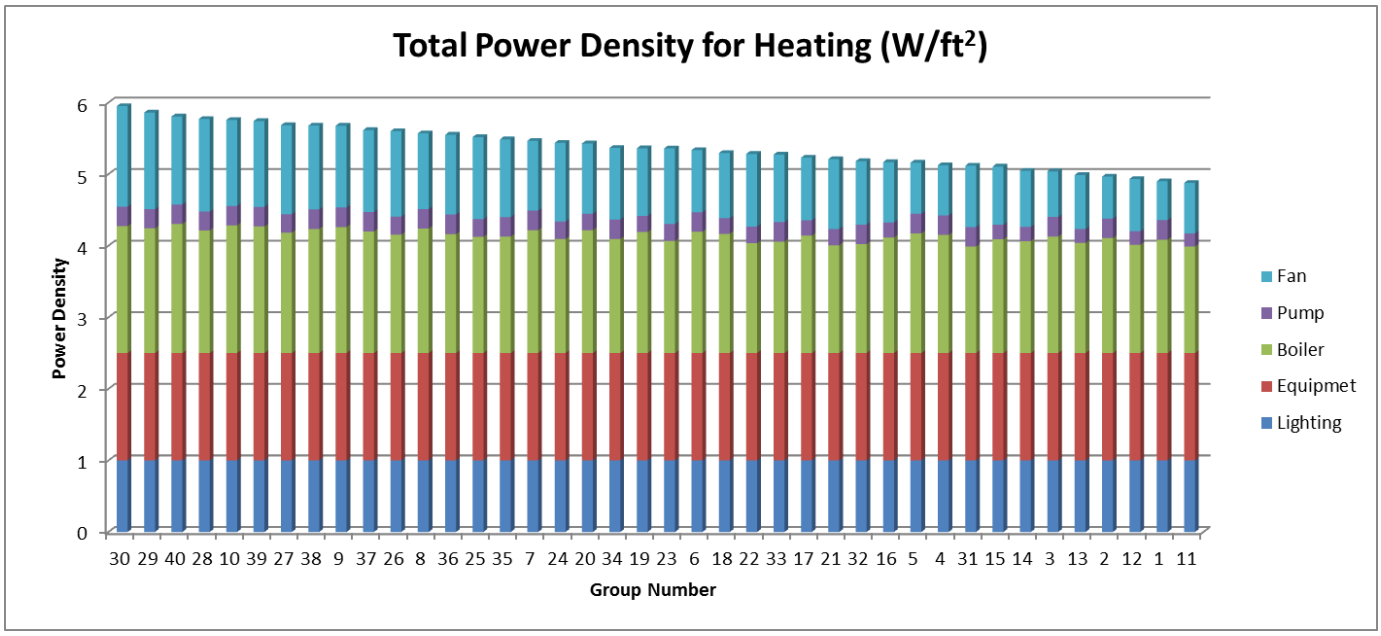


Figure 3. Total Power Density for Winter Condition

Design and Simulation of an Energy-Positive Building

Tiberiu Catalina, Razvan Popescu, Martha Soare, Ovidiu Serban, Nicolae Bajenaru
Technical University of Civil Engineering Bucharest
Faculty of Building Services Engineering, CAMBI Advanced Research Center
Bucharest, ROMANIA
tiberiu.catalina@gmail.com, razvan22@yahoo.com

Abstract—**This article presents the results from a simulation work regarding the analysis of energy balance of an positive energy building. To meet the criteria for an energy-positive house, a high priority was given to the performance of the thermal envelope, such as high insulation of walls, roofs, floors and windows, thermal bridge-free construction and air tightness. Due to the required air tightness, special attention was also paid to indoor air quality through proper ventilation. We have simulated the building and the heating/cooling system which in our case is a water source heat pump connected with U-pipes vertical boreholes. With the geothermal system, along with solar thermal system used to produce the domestic hot water demand and along with the photovoltaic-PV system we managed to obtain a positive energy construction. The multi-source system is simulated with multiple specialized software. The installed PV system produces 5379 kWh/year while the energy consumption of the house is 4856 kWh/year. During summer, in order to avoid overheating a set point of 27^{o}C was considered. For the entire simulation year, the interior temperature in all the zones was between 19^{o}C and 27^{o}C. This article presents the modeling and simulation of the multi-source systems and illustrates interesting insights about the right measures to obtain a high energy efficient house with optimum indoor comfort.**

Keywords-positive energy house; dynamic simulations; multi-source system.

I. INTRODUCTION

European Union (EU) has agreed a forward-looking political agenda to achieve its core energy objectives of sustainability, competitiveness and security of supply, by reducing greenhouse gas emissions by 20%, by increasing the share of renewable in the energy consumption to 20% and improving energy efficiency; all these, by 2020 [1].

The energy spent to heat the occupied spaces in the residential sector represents more than 40% from the total energy demand [2], which includes electricity, hot-water and air-conditioning. In this area, a major energy reduction can be achieved if a building is correctly designed by engineers and architects, and even more if, renewable energy systems are integrated to the construction. Installing multiple renewable sources on the same site is even more appealing when substantial energy savings could be made if the advantages of each source are associated. In the near future, more and more the renewable energy sources will cohabit with fossil energy source systems and research has to be pointed towards solutions that are energy efficiently, economical viable and environmental friendly. The goal of a multi-source system is to decrease at maximum the primary energy consumption by generating the needed demand by renewable sources like solar, wind or wood energy. The use of several sources on the same construction site will be applied for new, but also for buildings which, are on the way to be renovated. Coupling a heating system with a renewable energy system along with a multi-criteria decision analysis was realized by Catalina et al. [2]. The benefits of such a use is that the constructions can be closer to Zero Energy Buildings (ZEB) or even positive energy buildings since only by means of a multi-energy system we can arrive to such ambitious purpose. The renewable energy systems will produce locally the energy needed for the building and the extra energy, which is not necessary, will be sent to the overall urban energy infrastructure (i.e., the case of photovoltaic power energy or wind energy). A comparison of different ZEB in terms of thermal behavior was studied by Nazif and Altan [3]. The use of a ground to air heat exchanger for energy efficient houses in South of Europe was found to be an attracting solution in order to achieve a ZEB [4]. Compared to the other studies, in this article, a new multi-source system along with the design parameters to be taken into consideration for the envelope is presented. Moreover, if most of the studies are focused only on how to reduce the energy consumption, we found that first of all a building should provide a healthy and comfortable environment for the occupants. With this project it is showed that is possible to achieve a ZEB with a good comfort for the occupants. This is also the main advantage of the proposed system and approach. The structure of the article is divided several sections that are meant to present the study case building, the HVAC (Heating, Ventilation, Air-Conditioning) system, the results of the simulations and finally, the conclusions. During the next chapter we will present the building design of the study case, along with thermal modeling. Afterwards, it is shown the modeling of the HVAC system. At the end of the article are illustrated the results and the corresponding conclusions.

II. BUILDING DESIGN

The house selected for the study is located in Chambery, France. It is a detached two storey building and it is occupied all year long, with a difference between weekdays and weekends. The architectural plan of the building is illustrated

 ISBN: 978-1-61208-308-7

in Fig. 1. By its shape and space configuration, the house has to achieve, besides comfort and regular architectural image, the premises of an energy-positive house. Based on efficiency concept, the house is wrapped with a „thick skin", to allow an efficient thermal insulation and to have neglectable heat losses from thermal bridges. The envelope of the house will be plastered with the same material, in order to ensure air tightness.

A key factor in this house is the relation with the solar radiation, captured both directly (windows) and indirectly (solar collectors). Large windows are mainly oriented south, increasing the useful solar gains during winter and using efficiently the natural light during the year. Furthermore, the main areas are oriented south, east and west, while the small areas are oriented north side.

The house presents highly insulated facades and roof; in addition, the triple-glazed windows, with specially insulated frames, are based on a 6 chamber system, that keeps out draughts, dust and water. The total window U-value is 0,73 W/m^2K. The other elements of the house are presented in Table I.

TABLE I. HOUSE BUILDING MATERIALS AND INSULATION LEVELS

Type	Building structure materials and U-value
Exterior walls	Interior plaster (15 mm) POROTHERM 30 STh clay blocks (300 mm) Thermal insulation compound system (300 mm) out of polystyrene hard foam EPS, plastered on the outside **U=0.09W/m²K**
Interior walls	Interior plaster (15 mm) Thermokron 24 TK blocks (115 mm) Interior plaster (15 mm)
Roof	Clay tiles (20 mm) ISOVER VARIO KM membrane – vapour retarder and air tightness layer Mineral wool (350 mm) Plaster board (15 mm) **U=0.0977 W/m²K**
Ground floor	Oak wood flooring (30 mm) Concrete slab (200 mm) Rigid polyurethane foam (350 mm) Vapour retarder and air tightness layer Cement (50 mm) **U=0.094 W/m²K**
First floor	Concrete slab (100 mm) Interior plaster (15 mm) Oak wood flooring (30 mm)
Attic floor	Interior plaster (15 mm) Concrete slab (100 mm) Thermal insulation (400 mm) Concrete (50 mm) **U=0.0966 W/m²K**
Attic wall	Wood (100 mm) Thermal insulation (400 mm) Exterior plaster (20 mm) **U=0.0911 W/m²K**

The building has been introduced and modulated in TRNBuild, a component of TRNSYS 16 program [2]. The TRNBuild module allows users to define a number of building parameters including the orientation, envelope construction, glazing and infiltration rate. Once the building has been fully defined, it can be imported into the Simulation Studio to be linked with the weather file and HVAC system. We have used for the building the Type 56 component, a detailed multi-zone building model. This component models the thermal behavior of a building divided into different thermal zones.

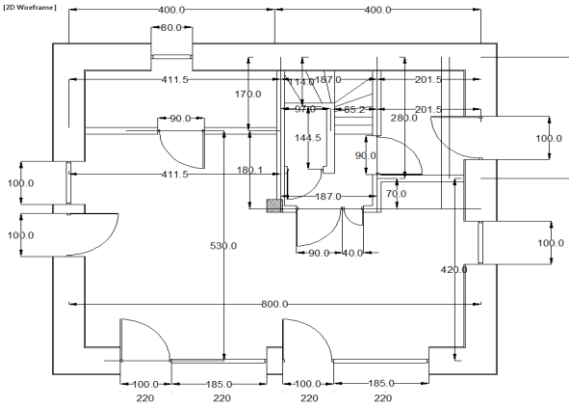


Figure 1. First floor plan of the energy positive house

The thermal zones are very important if we want to simulate the energy consumption of the building. These zones can include a room or several rooms that may have similar heat gains, similar profiles (temperature, occupancy, etc.) or they are provided by the same HVAC system. In our case, the house was divided in 10 different zones, presented in Table II.

TABLE II. THERMAL ZONES OF THE HOUSE

Zone	Space	S [m^2]	V [m^3]
1	Kitchen+Dining Room+Living Room	30,58	84,17
2	Hall	3,88	10,67
3	Storeroom	4,83	13,29
4	Staircase	5,3	15,36
5	Room 1	11,08	29,79
6	Room 2	10,83	29,1
7	Room 3	10,83	29,1
8	Bathroom	5,01	13,45
9	Attic	39,73	54,21
10	Sanitary void	46,81	49,15

The heat gains for people are very important and must be considered in the energy balance. They have a great influence on the energy consumption and the overheating of the rooms during the summer season. Occupancy profiles were created specific to each area and divided into periods of the week (weekdays and weekends). The power dissipation of a person is estimated to 100 W (60 W sensible heat and 40 W latent heat) for casual activities and the percentage of heat gains by convection and radiation to 33% radiation and 67% convection [3]. The artificial lighting is not constant and doesn't depend necessarily on the occupied area. The following control strategy was used: the lighting is on if horizontal solar radiation <120 W/m^2 and off if >200 W/m^2, taking into account the occupied area.

There is no need to heat the entire building day and night at 19°C, while the rooms are empty during the day and some areas are empty during the night. If the house is not

 ISBN: 978-1-61208-308-7

occupied, the temperature is set to 16°C. During the weekend, the temperature profile is linked to the occupancy profile in order to provide thermal comfort for every person.

III. HVAC SYSTEM MODELING

In recent years, the design requirements for the primary energy consumption of dwellings has been radically reduced, making it difficult, if not impossible, to meet the required levels using only a combination of construction measures and fossil fuel-based heating systems. This has set in motion a transition to alternative energy systems within domestic construction. Ventilation systems, heat pump systems and solar collector systems form an often-used alternative to provide heating and hot water in dwellings as they reduce the use of fossil energy sources.

The house has a ventilation system with built-in heat exchanger to recover heat, which can be operated by the occupants. There is no room which is not clearly integrated into the ventilation concept. The supply air is shared and areas with stagnant air do not exist. All living and sleeping rooms are planned as supply air zones, while the exhaust air rooms are the kitchen, storeroom and bathroom. The hallway and staircase act as overflow zones. The system is located in the building services room under the roof; supply and exhaust air are extracted or blown out directly above the roof.

A compact solution that combines the ventilation system and the space heating provides better efficiency and opportunities for different system solutions that can be adapted to different conditions and applications. The solution consists in linking a GSHP (Ground Source Heat Pump) to the heat recovery unit (see Fig. 2). A piping loop is buried in the ground, which is considerably warmer than the outdoor air in the winter. Water is circulated through the loops and into the building where the heat pump removes the heat from the water and delivers it to the air. The heat pump covers the whole heating demand with forced air heating, due to its distribution system, which transfers heat to the building. The process is reversed in cooling. Heat is removed from the inside air and delivered it to the water loop which rejects this heat to the ground. The GSHP also provides high cooling efficiency since the ground is much cooler than the air during the summer [4-8].

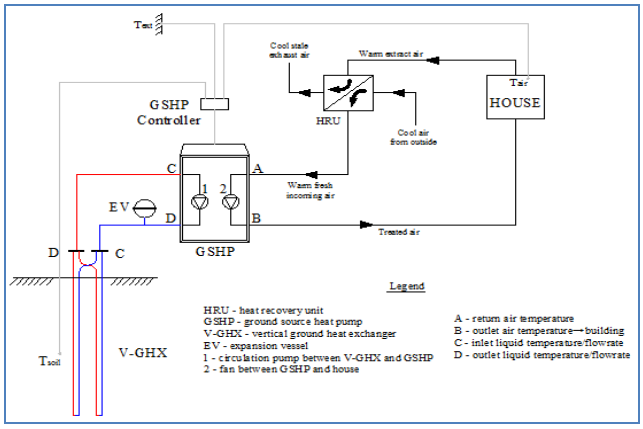


Figure 2. Ground source heat pump for heating/cooling

The production of domestic hot water is ensured by a thermal solar system, which provides heat through a solar collector. The collector transfers the heat to a storage tank, which is connected with an internal heat exchanger and a thermostatic valve for DHW (Domestic Hot Water) temperature control (see Fig 3). The STC (Solar Thermal Collector) is operating in series with the storage tank, which allows for an improvement in collector efficiencies due to the lower temperature fluid entering the collector. An efficient control strategy is implemented to regulate the system. The indoor temperature of the house is regulated by a thermostat, which controls the fan speed in the space heating circuit. Based on the temperature difference, the thermostat switches on/off or remains in its current state. The humidity control in the house is ensured by the GSHP's controller, which dehumidifies the space at reduced cooling capacity. The circulation pump in the solar circuit is controlled based on the temperature difference between the upper side of the STC and the bottom of the hot water tank. If the temperature difference is greater than (5-8)^{0}C, the pump is switched on, but when the difference is (1-3)^{0}C, the pump is switched off, due to the lack of heat that can be transferred to the tank. This difference can not drop below 1^{0}C, because the risk of cooling the tank will appear. The thermostat located in the upper area of the tank protects the equipment from temperatures higher than 90^{0}C. Otherwise, the pump is switched off and, for example, the pressure relief valve will be on. The production of on-site energy is ensured by a photovoltaic system. An analysis by numerical simulations of a system of photovoltaic modules is intended, in order to assess its potential to cover the electricity consumption of the house. The photovoltaic system was designed and analyzed with the PVSyst V6.0 program [12].

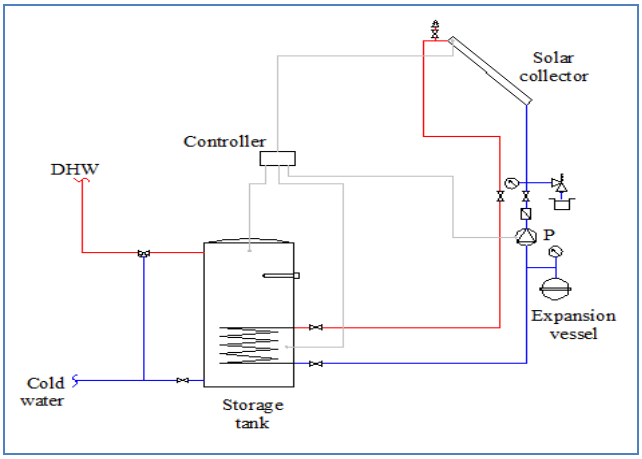


Figure 3. Domestic hot water production using solar panels

For the modeling and simulation of the entire system it was necessary the introduction of multiple parameters. The heat recovery has a sensible effectiveness of 0.95 and a power consumption of 500W. The heat pump is a water source based and provides a maximum air flow of 1250 m3/h

 ISBN: 978-1-61208-308-7

and a maximum heating production of 5.5 kW with a COP of 4.8 when the entering liquid temperature is 10 °C.

TABLE III. VERTICAL U-TUBES PROPRIETIES

Borehole depth	100 m
Number of boreholes	2
Header depth	2 m
Borehole radius	0,1016 m
Storage thermal conductivity	2,423 W/mK
Storage heat capacity	2016 kJ/m³/K
Outer radius of U-tube pipe	0,01664 m
Inner radius of U-tube pipe	0,01372 m
Fill thermal conductivity	8,722 kJ/hmK
Pipe thermal conductivity	1,5122 kJ/hmK
Reference borehole flow rate	1000 kg/h

The proprieties of the boreholes used for the simulations resumed in Table III.

IV. RESULTS

For an appropriate analysis of the system, a dynamic simulation is necessary. The numerical simulation is made throughout the year, for 8760 hours. The thermal comfort is acquired for the entire occupational period as the air temperature, during winter time, is not passing below 19°C and during summer period is always lower than 27°C. Fig. 4 illustrates the air temperature for the thermal zones 1 to 8. Zone 1 has higher temperatures, up to 22°C because in that zone we have the highest internal heat gains (kitchen, occupants, and other appliances). Zone 2 and Zone 3 with the lowest temperatures, close to 19°C represent the Entrance hall and the storage room. It can be concluded that the HVAC system is well designed and the indoor environment is comfortable.

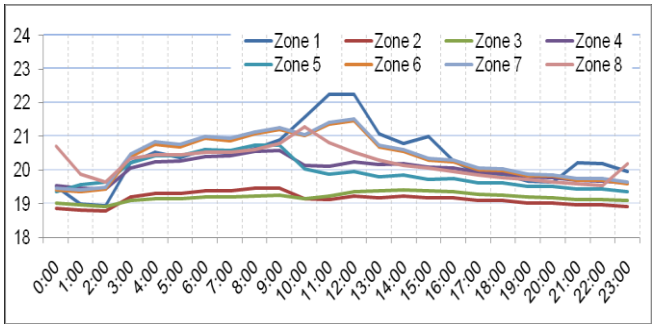


Figure 4. Air temperature during the coldest day of the year

The heat pump transfers to the introduced air a certain amount of energy, when it is needed. Figure 4 shows the total heat transfer to air and the heating controller during the coldest week of the year. The maximum value is around 5500 W and the lowest air temperatures are -12°C. As it can be noticed from Fig. 5, the functioning hours are low because of the high level of insulation of the house and of the internal heat gains that covers most of the heating demand.

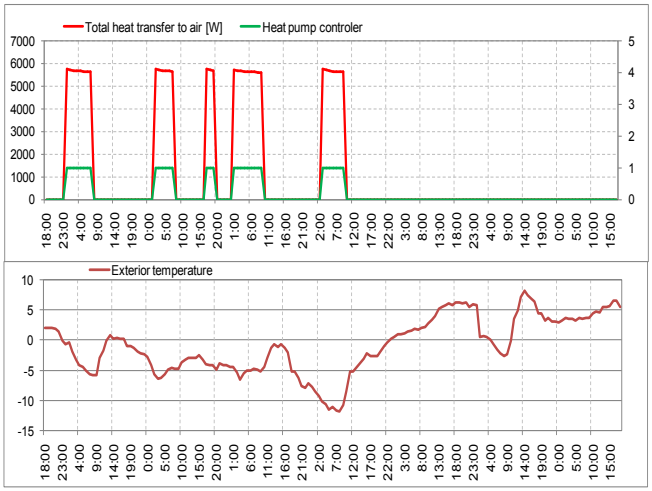


Figure 5. GSHP functioning during the coldest week of the year

As concerns the cooling energy demand, during the warmest week of the year the heat pump transfers to the introduced air around 7000 W (see Fig. 6). This energy is required to avoid the overheating of the zones. Compared to the winter situation, it can be clearly noticed that the functioning hours of the GSHP are more. The reason for that is because the solar radiation heat gains and internal heat gains have high values. Moreover, the outdoor air temperatures during the summer period are around 35°C.

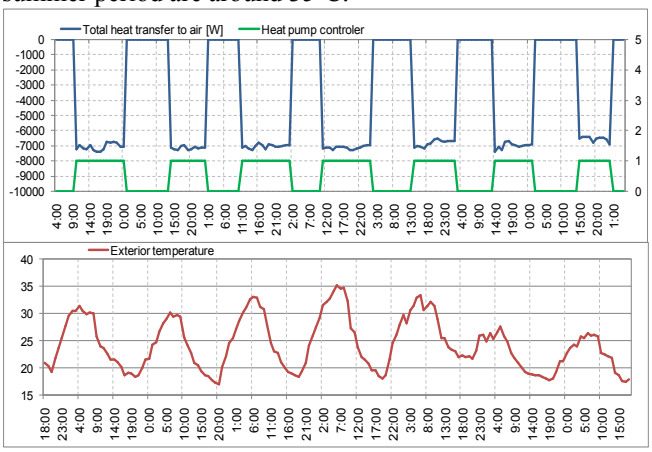
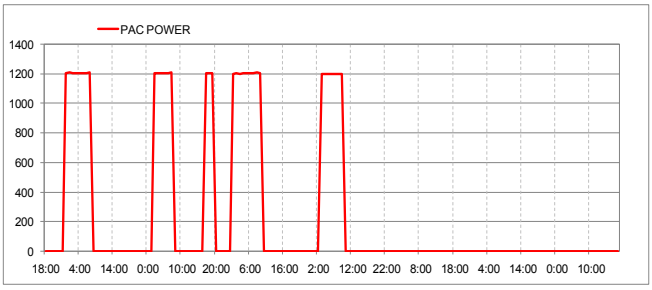


Figure 6. GSHP functioning during the warmest week of the year



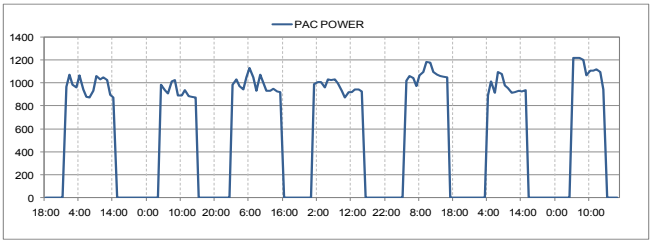


Figure 7. GSHP power consumption during the coldest/warmest week

The power consumption of the heat pump is important for an energy efficient building. From Fig. 7, it can be noticed that during the winter/summer period the maximum needed energy is 1200 W. This energy consumption comprises the compressor, the controller and the blower. For the entire year period we have a consumption of 1853 kWh.

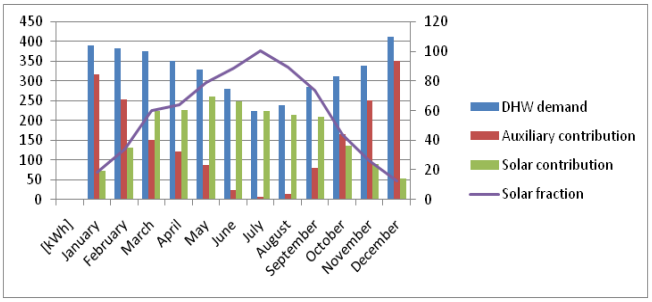


Figure 8. Energy balance of the DHW solar production

The DHW energetic demand is 3932 kWh/year. As it can be observed in Fig. 8, the largest amount of energy from solar contribution is provided during May, when the value of 250 kWh/month is exceeded. The lowest amounts of solar contribution are during the months when outdoor temperatures and solar radiation intensity have lower values. The solar energy is not enough to cover the DHW energetic demand; therefore an auxiliary heating device is necessary. The auxiliary contribution covers entirely the demand and the value is 1833 kWh.

As concerns the photovoltaic system this one is oriented south and is located on the roof of the house. The panels are polycrystalline silicon cells, for the best performance. The input data are: tilting angle: 30°; Azimuth: 0°; Number of PV modules: 17; The module power: 0.255 kWp, The total PV system power: 6 kWp; Module type: Polycrystalline; Modules efficiency: 15.7%. The energy produced by the PV system is 5266 kWh. The necessary energy for the appliances is estimated at 2.66 kWh/day = 971 kWh/year and the electrical lighting at 200 kWh/year. The energy balance of the house is positive because we have 1853 kWh (Heat pump) + 1833 kWh (Auxiliary heating DHW) + 970 kWh (Appliances) + 200 kWh (Electrical lighting) = 4856 kWh (Total consumption) and the produced energy is 5372.9 kWh.

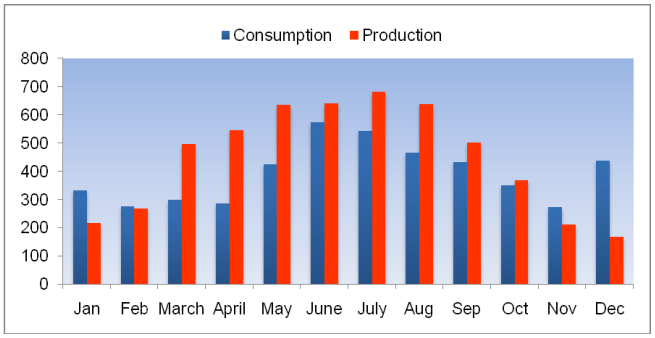


Figure 9. Energy production of the PV system and energy consumption

The maximum energy consumption is found in June with 580 kWh, while the maximum production is July with 685 kWh (see Fig. 9).

V. CONCLUSIONS

In this study, we analyzed a single family house and the possibility to obtain an energy positive balance. Using a ground source heat pump the indoor comfort conditions were reached while the energy consumption had low values. A solar thermal system coupled with an auxiliary heating system produced the domestic hot water demand of the four occupants. An energy positive house would not have been possible without the use of photovoltaic panels that is why 17 modules of 0.255 kWhp/module were installed. With this amount of PV panels the energy production was of 5372.9 kWh over passing the energy consumption by 516.9 kWh. It is concluded that the proposed multi-source system was correctly designed and that the simulations were the perfect way to analyze the house and the HVAC system.

ACKNOWLEDGMENT

This work was supported by a grant of the Romanian National Authority for Scientific Research, CNCS – UEFISCDI, project number PN-II-RU-TE-2012-3-0108.

REFERENCES

[1] Memo: EU energy security and solidarity action plan: 2nd strategic energy review, Online reference : http://ec.europa.eu/energy/strategies/2008.
Retrieved: February , 2011

[2] Catalina T., Virgone J., and Blanco E., Multi-source energy systems analysis using a multi-criteria decision aid methodology, Renewable Energy, Volume 36, Issue 8, August 2011, pp. 2245-2252.

[3] Nazif G. and Altan H., Zero energy house design for cyprus: enhancing energy efficiency with vernacular techniques, Proceedings of Building Simulation 2013 conference, August 2013, pp. 2978-2984.

[4] Pagliano L. and Zangheri P., Design of nearly zero energy buildings coupled with an earth to air heat exchanger in mediterranean climate: development of an analytic model and validation against a monitored case study, Proceedings of Building Simulation 2013 conference, August 2013, pp. 3705-3711.

 ISBN: 978-1-61208-308-7

[5] Trnsys 16, A Transient System Simulation Program, Solar Energy Laboratory, 2005, University of Wisconsin Madison, USA.

[6] ASHRAE Handbook, HVAC Applications, American Society of Heating, Refrigeration and Air-Conditioning Engineers, Inc., Atlanta, GA, 1995.

[7] S. Kavanaugh and K. Rafferty, Ground-Source Heat Pumps, ASHRAE Transactions, 1997.

[8] J.E. Bose, M.D. Smith, and J.D. Spitler, Advances in ground source heat pump systems an international overview, Proceedings of the Seventh International Energy Agency Heat Pump Conference, May 19-22 2002, Beijing, pp. 313-324.

[9] J.D. Spitler, S.J. Rees, and C. Yavuzturk, Recent Developments in Ground Source Heat Pump System Design: Modelling and Applications. Proceedings of CIBSE/ASHRAE conference, Dublin, September 2000, Session 9a, Paper A28, pp. 1-34.

[10] IGSHPA, Closed-Loop/Ground-Source Heat Pump Systems – Installation Guide, International Ground-source Heat Pump Association, Oklahoma State University, Stillwater, Oklahoma, USA, 1988.

[11] P.K. Kavanaugh and K. Rafferty, Ground-source Heat Pumps – Design of Geothermal Systems For Commercial and Institutional Buildings, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., 1791 Tullie Circle, N.E., Atlanta, GA, USA, 1997.

[12] Online reference : http://www.pvsyst.com/en/, Retrieved: October, 2013

 ISBN: 978-1-61208-308-7