



SIMUL 2015

The Seventh International Conference on Advances in System Simulation

ISBN: 978-1-61208-442-8

November 15 - 20, 2015

Barcelona, Spain

SIMUL 2015 Editors

Philipp Helle, Airbus Group Innovations, Germany

Mario Freire, University of Beira Interior, Portugal

SIMUL 2015

Forward

The Seventh International Conference on Advances in System Simulation (SIMUL 2015), held on November 15 - 20, 2015 in Barcelona, Spain,, 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 2015 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 2015 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 2015. 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 2015 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 2015 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 Barcelona provided a pleasant environment during the conference and everyone saved some time for exploring this beautiful city.

SIMUL 2015 Advisory Chairs

Christoph Reinhart, Harvard University - Cambridge, USA
Amr Arisha, College of Business, DIT, Ireland

SIMUL 2015 Research Liaison Chairs

Tae-Eog Lee, KAIST, Korea

Marko Jaakola, VTT Technical Research Centre of Finland, Finland

SIMUL 2015 Industry Liaison Chairs

Shengnan Wu, American Airlines, USA

Tejas R. Gandhi, Virtua Health-Marlton, USA

SIMUL 2015 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 2015 Publicity Chairs

Nuno Melao, Catholic University of Portugal - Viseu, Portugal

SIMUL 2015

Committee

SIMUL Advisory Chairs

Christoph Reinhart, Harvard University - Cambridge, USA
Amr Arisha, College of Business, DIT, Ireland

SIMUL 2015 Research Liaison Chairs

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

SIMUL 2015 Industry Liaison Chairs

Shengnan Wu, American Airlines, USA
Tejas R. Gandhi, Virtua Health-Marlton, USA

SIMUL 2015 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 2015 Publicity Chairs

Nuno Melão, Polytechnic Institute of Viseu, Portugal

SIMUL 2015 Technical Program Committee

Kareem Abdelgawadm, Heinz Nixdorf Institute - University of Paderborn, Germany
Erika Abraham, RWTH Aachen University, Germany
Carole Adam, University Grenoble-Alpes, France
Petra Ahrweiler, EA European Academy of Technology and Innovation Assessment GmbH, Germany
Dimosthenis Anagnostopoulos, Harokopio University of Athens, Greece
Chrissanthi Angeli, Technological Institute of Piraeus - Athens, Greece
Amr Arisha, College of Business - DIT, Ireland
Enrico Barbierato, Seconda Università degli Studi di Napoli, Italy
Marek Bauer, Politechnika Krakowska, Poland
Ateet Bhalla, Independent Consultant, India

Kashif Bilal, COMSATS Institute of Information Technology, Pakistan
Keith Bisset, Virginia Tech, USA
Georgiy Bobashev, RTI International -Research Triangle Park, USA
Christophe Bourdin, Université d'Aix-Marseille, France
Jan F. Broenink, University of Twente, Netherlands
Dilay Celebi, Istanbul Technical University, Turkey
Srinivas R. Chakravarthy, Kettering University, USA
E Jack Chen, BASF Corporation, USA
Soolyeon Cho, North Carolina State University - Raleigh, USA
Franco Cicirelli, Università della Calabria, Italy
Kendra Cooper, University of Texas at Dallas, USA
Duilio Curcio, University of Calabria - Rende (CS), Italy
Andrea D'Ambrogio, University of Roma "Tor Vergata", Italy
Yuya Dan, Matsuyama University, Japan
Saber Darmoul, King Saud University, Saudi Arabia
Jacinto Dávila, Universidad de Los Andes, Venezuela
Luis Antonio de Santa-Eulalia, Université de Sherbrooke, Canada
Luís de Sousa, Public Research Centre Henri Tudor - Luxembourg, Luxembourg
Tom Dhaene, Ghent University - IBBT, Belgium
Jan Dijkstra, Eindhoven University of Technology, Netherlands
Atakan Dogan, Anadolu University, Turkey
Julie Dugdale, Université Pierre Mendès, France
Khaled S. El-Kilany, Arab Academy for Science - Alexandria, Egypt
Sabeur Elkosantini, Higher Institute of Computer Science of Mahdia - University of Monatir, Tunisia
Zhou Fang, VR and Immersive Simulation Center - Renault DE-TD, France
Francisco Javier Otamendi Fernández de la Puebla, Universidad Rey Juan Carlos, Spain
Ian Flood, University of Florida, USA
Martin Fraenzle, Carl von Ossietzky Universität Oldenburg, Germany
Terrill L. Frantz, Peking University HSBC Business School, Shenzhen Campus, Guangdong, China
Jason Friedman, Tel Aviv University, Israel
José Manuel Galán, Universidad de Burgos, Spain
Tejas Gandhi, Medical Center of Central Georgia, USA
Hector Miguel Gastelum Gonzalez, University of Guadalajara, Mexico
Petia Georgieva, University of Aveiro, Portugal
Genady Grabarnik, St. John's University, USA
Antoni Grau, Technical University of Catalonia UPC, Spain
Francisco Grimaldo Moreno, Universitat de València, Spain
Christoph Grimm, TU Kaiserslautern, Germany
Amine Hamri, LSIS Marseille, France
Samer Hassan, Universidad Complutense de Madrid, Spain
Frank Herrmann, Regensburg Technical University of Applied Sciences, Germany
Tsan-sheng Hsu, Academia Sinica, Taiwan
Xiaolin Hu, Georgia State University, USA
Mauro Iacono, Seconda Università degli Studi di Napoli, Italy
Eric Innocenti, University of Corsica Pasquale Paoli, France
Marko Jaakola, VTT Technical Research Centre of Finland, Finland
Cecilia Jaramillo, University Autònoma of Barcelona (UAB), Spain
András Jávör, Budapest University of Technology and Economics, Hungary

Emilio Jiménez Macías, University of La Rioja, Spain
Christina Kluever, University of Duisburg-Essen, Germany
Natallia Kokash, Centrum Wiskunde & Informatica (CWI), Netherlands
Sunil Kothari, HP Labs, USA
Olexandr O. Kuzenkov, Khmelnytsky National University, Ukraine
Pierre L'Ecuyer, Université de Montréal, Canada
SangHyun Lee, University of Michigan, USA
Fedor Lehocki, National Centre of Telemedicine Services / Slovak University of Technology in Bratislava, Slovakia
Jennie Lioris, CERMICS, France
Rong Liu, University of California, Los Angeles, USA
Francesco Longo, University of Calabria, Italy
Jose Machado, Universidade do Minho, Portugal
Michael Manitz, Universität Duisburg/Essen, Germany
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
Don McNickle, University of Canterbury - Christchurch, New Zealand
Nuno Melão, Polytechnic Institute of Viseu, Portugal
Jürgen Melzner, Bauhaus-University Weimar, Germany
Marco Mevius, HTWG Konstanz, Germany
Adel Mhamdi, RWTH Aachen University, Germany
Bożena Mielczarek, Wrocław University of Technology, Poland
Saurabh Mittal, Dunip Technologies, USA
Lars Mönch, University of Hagen, Germany
Roberto Montemanni, University of Applied Sciences of Southern Switzerland (SUPSI), Switzerland
Àngela Nebot, Polytechnic University of Catalonia, Spain
Liberio Nigro, Università della Calabria, Italy
Lialia Nikitina, Fraunhofer Institute for Algorithms and Scientific Computing, Germany
Mara Nikolaidou, Harokopio University of Athens, Greece
Aida Omerovic, SINTEF ICT, Norway
Tuncer Ören, University of Ottawa, Canada
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
Tomas Potuzak, University of West Bohemia, Czech Republic
Francesco Quaglia, Sapienza Università di Roma, Italy
Arash Ramezani, University of the Federal Armed Forces, Germany
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

Hendrik Rothe, Helmut Schmidt Universitat, Germany
Juliette Rouchier, GREQAM - CNRS, France
Manuel Filipe Santos, University of Minho, Portugal
Jean-Francois Santucci, University of Corsica, France
Guodong Shao, National Institute of Standards and Technology - Gaithersburg, USA
Yuri N. Skiba, Universidad Nacional Autónoma de México, Mexico
Jeffrey S. Smith, Auburn University, USA
Flavio Soares Correa da Silva, University of Sao Paulo, Brazil
Eric Solano, RTI International, USA
Yuri N. Sotskov, United Institute of Informatics Problems - National Academy of Sciences of Belarus, Belarus
Nary Subramanian, University of Texas at Tyler, USA
Changho Sung, Korea Institute of Science and Technology Europe (KIST Europe), Germany
Magdalena Szmajduch, Cracow University of Technology, Poland
Elena Tànfani, University of Genova, Italy
Alexander Tatashev, Moscow University of Communications and Informatics, Russia
Pietro Terna, University of Torino, Italy
Georgios Theodoropoulos, Durham University, UK
Michele Tizzoni, Institute for Scientific Interchange, Torino, Italy
Klaus G. Troitzsch, Universität Koblenz-Landau, Germany
Kay Tucci, Universidad de Los Andes, Venezuela
Alfonso Urquia, Dept. Informatica y Automatica - UNED, Spain
Manuel Villen-Altamirano, Universidad de Malaga, Spain
Antonio Viridis, University of Pisa, Italy
Shengyong Wang, The University of Akron, USA
Frank Werner, Otto-von-Guericke-University Magdeburg, Germany
Andi Widok, HTW Berlin (University of Applied Sciences), Germany
Philip Wilsey, Experimental Computing Lab - University of Cincinnati, USA
Kuan Yew Wong, Universiti Teknologi Malaysia, Malaysia
Shengnan Wu, Capital One Financial Corp., USA
Nong Ye, Arizona State University, USA
Levent Yilmaz, Auburn University, USA
Yao Yiping, National University of Defence Technology - Hunan, China
Gregory Zacharewicz, Universite de Bordeaux, France
Justyna Zander, Berner & Mattner Systemtechnik GmbH, Germany
František Zboril, Brno University of Technology, Czech Republic
Ouarda Zedadra, Laboratory of Science and Technology of Information and Communication (LabSTIC), Algeria
Liu Zhengchun, University Autonoma of Barcelona (UAB), Spain
Armin Zimmermann, Technische Universität Ilmenau, Germany

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 to 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 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

Simulation Aids Long-Term Capacity Planning at a Plant Manufacturing Safes <i>Swapnil Landge and Edward J. Williams</i>	1
Efficient Modelling and Real-Time Optimisation of Stationary Systems: Application to an Evaporation Process <i>Carlos Gomez Palacin, Jose Luis Pitarch, and Cesar De Prada</i>	6
An Analytical Model to Evaluate the Response Capacity of Emergency Departments in Extreme Situations <i>Eva Bruballa, Manel Taboada, Alvaro Wong, Dolores Rexachs, and Emilio Luque</i>	12
Differentiated Service Queuing Disciplines in NS-3 <i>Robert Chang, Mahdi Rahimi, and Vahab Pournaghshband</i>	17
Simulation of the Influence of Increased Speed Limit and Densification of Stations on Travel Time Based on the Example of the First Metro Line in Warsaw <i>Marek Bauer</i>	24
Upgrade of the MovSim for Easy Traffic Network Modification <i>Andraz Omahen, Teo Kukuljan, Vida Vukasinovic, Peter Korosec, and Gregor Papa</i>	31
Evaluation Method for Appropriate Fleet Size of Forklifts at a Freight Station <i>Shunichi Tanaka</i>	37
Statistical Emulation Applied to a Very Large Data Set Generated by an Agent-based Model <i>Wim De Mulder, Bernhard Rengs, Geert Molenberghs, Thomas Fent, and Geert Verbeke</i>	43
Using Simulation Modeling to Create Transdisciplinary Process Models and to Build Basic PLM Tools to Support the Systems Engineering Lifecycle <i>Germano S. Kienbaum, Renato Fernandez, Alvaro A. Neto, and Elaino K. T. Silva</i>	49
Simulation Methodologies for the Numerical Analysis of High-Speed Dynamics <i>Arash Ramezani and Hendrik Rothe</i>	59
Modeling I/O System in HPC: An ABMS Approach <i>Diego Encinas, Marcelo Naiouf, Armando De Giusti, Sandra Mendez, Dolores Rexachs, and Emilio Luque</i>	67
Using a Generic Modular Mapping Framework for Simulation Model Composition <i>Philipp Helle and Wladimir Schamai</i>	72
Memory Constrained Iterative Phase Retrieval using GPGPU <i>Ladislav Mikes</i>	79

Resource Management and Optimization Efficiency Issues in a Simulation Software Package <i>Havatzelet Cohen, Daniel Khankin, Shlomo Mark, and Shaul Mordechai</i>	84
Analytical Modeling of Parametric Yield Considering Variations in Leakage Power and Performance of Nano-Scaled Integrated Circuits <i>Hossein Aghababa, Nima Mozaffari, and Behjat Forouzandeh</i>	89
Combining Finite Element and Multibody Modeling Techniques for Time-Efficient Simulation of Nonlinear Gear Dynamics <i>Jakub Korta, Antonio Palermo, Domenico Mundo, and Shadi Shweiki</i>	94
Alternative System Design Based on Reliability Analysis and Simulation <i>Merve Uzuner and Berna Dengiz</i>	100

Simulation Aids Long-Term Capacity Planning at a Sunglasses Manufacturing Plant

Swapnil Landge

Industrial and Systems Engineering
North Carolina State University
Raleigh, North Carolina 27695 United States
sjlandge@ncsu.edu

Edward J. Williams

College of Business
University of Michigan - Dearborn
Dearborn, Michigan 48126 United States
williams@umich.edu

Abstract— Discrete-event process simulation has long been able to analyze knotty problems arising in manufacturing, warehousing, health care, transportation (rail, air, bus, etc.), and service industries such as banks, restaurants, and hotels. These knotty problems include challenges such as reducing inventory, increasing production (throughput), deploying workers efficiently, and reducing both lengths of queues and time spent in those queues. Indeed, from a historical perspective, the first, and still some of the most conspicuous, successes of simulation have been achieved in its applications to manufacturing. The application of simulation described in this paper arose in the context of manufacturing safes from their raw-material shells. Simulation, in contrast to other methods such as closed-form optimization, is highly capable of accommodating high process variability and almost automatically providing “best-case” and “worst-case” (as well as averages) for important performance metrics such as lengths of queues and waiting times in queues. Additionally, the animation which routinely accompanies simulation helps non-technical managers understand the results. In this context, the most painfully pressing problem was excess inventory, coupled with too slow and too meager output. The simulation study guided engineers and managers as they endeavored to both reduce the inventory and increase the rate of output – only very rarely can these two objectives be achieved concurrently.

Keywords-discrete-event process simulation, manufacturing, capacity planning, throughput, queuing analysis, inventory control.

I. INTRODUCTION

In the roughly half-century since discrete-event process simulation made the transition from research curiosity to vital business-analytics tool, the earliest and also some of its most impressive successes have occurred in its applications to manufacturing processes (Law and McComas [1]). For example, (Zülch and Zülch [2]) applied simulation to the design of hybrid U-shaped assembly systems, an NP-hard problem of assigning operations to stations within an assembly line. Simulation effectively established priority rules for scheduling of a flow with simultaneously loaded stations, as documented by Hermann ([3]). A hybrid approach of a genetic algorithm and simulation produced excellent results in a job-sequencing problem within a semi-

automated production process, as described by Mosca, Queirolo, and Tonelli [4].

The present application of simulation is to a semi-automated manufacturing line producing safes (strong reinforced metal boxes, resistant to burglary, for the secure storage of relatively small valuables (e.g., jewelry, passports, stock certificates, etc.). Severe economic challenges besetting this process included excessive raw material inventory (expensive in both space and time), insufficient production output achieved too slowly, and inefficient deployment of workers resulting in both conspicuous idle time and sporadic lack of a worker needed to perform a task of high urgency, such as repair of a malfunctioning machine. Discrete-event process simulation, unlike many other analytical techniques such as closed-form optimization, has several significant advantages which made it highly suitable for attacking these challenges:

1. Ability to routinely accommodate high process variability via appropriate use of specified suitable probability distributions
2. Ability to provide extreme values, in addition to expected values, for key performance metrics such as waiting time in queue, length of queue, time-in-system, and output per working day
3. Ability to provide, routinely and with nearly zero incremental effort, an animation which greatly helped non-technical management personnel understand and accept the results provided by the analysis.

It has often been well and truly said that “simulation is like a movie, not like a still photograph.”

The rest of this paper is structured as follows: Section II presents an overview of the manufacturing process and Section III a description of data collection. Section IV describes the construction, verification, and validation of the model. Section V presents its results. In Section VI we present conclusions and indicate likely directions of future work.

II. OVERVIEW OF THE MANUFACTURING PROCESS

The manufacturing process which constructs safes from raw materials is a semi-automated assembly line comprising a total of eighteen workstations, nine of which are automated and nine of which are manual. The manual workstations are

each operated by two workers. All eighteen workstations have idiosyncratic cycle times and downtime performances (time to failure and time to repair). The line has limited space for incoming raw materials immediately upstream from the first workstation and a buffer of capacity three, for finished product, just downstream from the last workstation. Workflow is entirely linear along all eighteen workstations, with zero buffer capacity in the seventeen transits between the workstations. Furthermore, material movement is done by an indexed conveyor, as described by Gunal, Sadakane, and Williams [5]; that is to say, no part movement can be undertaken until:

1. All eighteen workstations have completed their processing cycle, *and*
2. All workstations are ready to accept a new part (i.e., are not “down.”)

Each workstation sends a “job over” signal when it has satisfactorily completed its cycle and hence is willing to “let” the conveyor index. Only when that signal has been received from all workstations can the conveyor index (move each work-in-process item forward one workstation).

Raw material arrives from a warehouse via a truck able to carry twenty shells. These shells are measured against tolerances and reworked if necessary before proceeding to the first workstation (reworked shells have priority over newly arriving shells). As they proceed to the first workstation via forklift, these shells are grouped with other components (lugs, stiffeners, and joint strips). Due to in-process space constraints, the forklift transfers shells only when the number of shells already waiting at the first workstation falls below six. This same forklift transfers finished safes from the last workstation to an in-plant storage location. The tasks to be done manually at the nine manual workstations are sufficiently dissimilar that workers (who are not cross-trained) are constrained to work only at their designated workstation.

Factory managers were keenly aware of process deficiencies and highly eager to eliminate or mitigate them, but also wished to proceed circumspectly due to the following downsides:

1. Experimental revisions to the process would entail lost production time.
2. If a proposed improvement involved rearranging machines, and then failed to live up to its promise, returning the machines to their original locations would be costly in both time and money.
3. Upper-level financial managers naturally wanted to see strong evidence of expected improvements before investing corporate funds.
4. In the absence of analytical tools, any heuristic cost-benefit analyses of proposed changes would be frustratingly vague.

As it so often has in the past, discrete-event process simulation, by virtue of allowing the actual system to continue operation while proposed improvements to it are studied via analysis of a model, provided an attractive circumvention of this seeming impasse.

III. INPUT DATA AND ITS ANALYSIS

The automatic workstations have essentially constant cycle times. The manual ones do not; for each of them, actual cycle times were collected via deliberately unobtrusive observation (beware the Hawthorne effect, as cautioned by Kroemen and Grandjean [6]). These data were then fitted to theoretical closed-form distributions using distribution-fitting software. The techniques of using such software have been documented by Chung [7], and the specific software used, Stat:Fit® is described by Leemis [8]. For each of the nine manual workstations, the algorithms in this software (Anderson-Darling, Kolmogorov-Smirnov, and chi-squared) recommended use of a uniform distribution. Likewise, times between failures and times to repair were fitted to all eighteen workstations using exponential and PERT distributions respectively. The PERT distribution, like the triangular distribution, has minimum, mode, and maximum parameters, but also two advantages over the triangular: it is differentiable throughout the interior of its range, and it has less probability mass in its tails – the converse of the latter being a criticism frequently directed against indiscriminate use of the triangular distribution.

Other needed data were readily available. The carrying capacity of the supply truck, and its travel time between the warehouse and the assembly line, were readily observable; the same was true for the forklift travel times and capacity. Experience indicated 10% of the shells must be sent to rework before entering the processing line. Furthermore, the shift schedules, including break times, followed by the assembly-line workers and the three mechanics of the maintenance department (required to repair failed workstations) were known.

IV. MODEL CONSTRUCTION, VERIFICATION, AND VALIDATION

Simio® simulation software, thoroughly documented by Thiesing and Pegden ([9]) and Kelton, Smith, and Sturrock ([10]) was used to build the simulation model of the manufacturing plant. This simulation software tool, both powerful and easy to learn and use, provides canonical constructs for entities, material-handling vehicles, conveyors, workstations, and workers. In the model, in accordance with current practice, shells arrive from the upstream plant via truck to storage, from whence they are moved to the first workstation by forklift. The shells then move sequentially among the eighteen workstations via the indexed conveyor. Then the truck returns the finished safes to the originating plant. The model also includes downtimes and repair operations (undertaken by specialized workers) at the workstations, plus workstation changeover times required when a new type of shell is about to enter the production line. Currently, such a changeover is restricted to occur only at the start of a new work shift, with the changeover being done during the scheduled time between successive work shifts. Simio® allows the modeler to represent process logic such as this in a “drag-&-drop” flowchart, shown in Figure 1 (Appendix). Additional examples of Simio® modeling constructs which proved very useful in this model were

“Material,” to conveniently track raw-materials usage and needs for replenishment, and “Monitor,” which can trigger appropriate logic within the model when the value of a state variable (e.g., an inventory level) crosses a certain threshold value in a specialized direction (downward, upward, or both).

Verification was undertaken first; then validation was undertaken. Techniques described by Hagan ([11]) were used; these techniques included structured walkthroughs, step-by-step examination of the animation (which Simio® automatically built as the simulation model was built), and close monitoring of the output metrics: queue lengths and inventory levels in the model versus those observed in practice, percentage of time workstations were idle waiting for other workstations to complete their cycle so the conveyor could index, frequency of trips made by the truck and the forklift, and utilization levels of the mechanics assigned to repair malfunctioning workstations. After adjustments to the model and correction of errors, the final model coordinated to 5% tolerance with system observations and historical data.

V. RESULTS OF THE SIMULATION MODEL

After completion of verification and validation, the model representing the current system was run for 20 replications of 24 hours each. Results agreed with currently observed values of performance metrics within 4%. At this point, the client managers accepted the model as valid and credible, opening the door to evaluation of one or more potential improvements. Both managers and analysts, based on extensive industry experience, were cognizant of the possibility of synergy: “Change A may produce negligible improvement; change B may produce negligible improvement, yet change A+B may produce significant improvement.”

To investigate various potentials for improvement, Simio® (and many other simulation software tools similarly) provides an Experiment option permitting concurrent evaluation of many Scenarios. In each Scenario, different values for model parameters (ranging widely among, for example, downtime frequency, downtime duration, buffer sizes, numbers of workers, cycle times, operational policy changes, etc.) may be specified. The multiple Scenarios are then run on a “one-click” basis and specified performance metrics (e.g., average and maximum length of queue and/or time in queue) easily compared via automatically generated graphs and tables. This approach proved both more flexible and quicker to implement than the perhaps more traditional “define a fitness function and run an optimization loop.”

Having already noticed (1) the low utilization of the delivery truck (recall it is responsible both for bringing shells from the warehouse to the production line being modeled and also for carrying completed safes back to the warehouse) and (2) chronically high work-in-process [WIP] levels, the first potential improvement modeled was “have the truck run twice as frequently with half the load sizes” – loads were reduced from 20 to 10 in both directions. For this first attempt at improvement, the carrying capacity of the forklift

remained at 3. The following “before & after” improvements were observed:

TABLE 1. SUMMARY OF FIRST IMPROVEMENT ATTEMPT

Performance Metric	Before Load Reduction	After Load Reduction
Average incoming shells in queue	5.69±0.14	3.99±0.12
Average outgoing safes in queue	6.93±0.20	3.76±0.06
Type 1 safes produced	58.9±4.08	69.7±2.21
Type 2 safes produced	26.10±4.30	45.8±2.29
Utilization of forklift	64.93±1.13%	57.04±0.50%
Utilization of truck	8.19±0.29%	20.80±0.36%

The confidence level for these intervals is 95%; note that no two of the before-versus-after intervals, considered pairwise, overlap.

The second step toward improvement stemmed from the observation that the three maintenance workers (both in observed practice and in the runs of the model made thus far) had very low utilizations. Therefore, the model was run with the reduced loads of 10 shown above and only one maintenance worker instead of three. In view of the low worker utilizations, this revision of the model was run with replications of length 500 hours, versus 24 hours. Utilizations of the three workers were 2.2%, 2.67%, and 2.58%; when only one worker was allocated to maintenance work, (1) The significant improvements achieved by the reduction in truck load size (more than 20% for both types of safes) were maintained, and (2) The single worker’s utilization remained extremely low at 7.86%.

Neither of these enhancements required any capital investment; indeed, the second one actually reduced staffing requirements. As the next and third step forward, the client managers and simulation analysts noted that whereas the current assembly line was completely linear, it could potentially be reconfigured in a “U” shape. This reconfiguration would surely entail expense, but held two enticements whose generic attractiveness has been confirmed by Groover ([12]):

The distance traveled by the forklift between the last workstation and the truck, when carrying completed safes, would be reduced from 30 meters to 7 meters (of relatively minor importance).

Three workstation pairs -- 8 and 12; 6 and 14; and 4 and 15 would be much closer together, making it practical to cross-train those pairs of assembly-line workers (of major importance).

A model animation snapshot of this revised layout of the assembly line appears in the Appendix (Figure 2).

This new scenario presented an interesting modeling challenge readily handled by Simio® logical expressions incorporated into the model logic. Specifically, the challenge can be characterized as follows: Suppose each of two workers, A and B, are busy on a task. Working alone,

worker A will need x minutes, and has already worked y minutes ($y < x$) when worker B finishes his task on another machine and joins worker A. Worker A's remaining time now decreases from $(x - y)$ to $(x - y) / 2$. Indeed, observation of the actual work undertaken at the manual workstations, plus discussion with the client managers, supported the assumption (underlying this computation) that the participation of a second worker involves negligible overlap or redundancy of work. In this scenario, the following improvements appeared:

1. Worker 4 utilization increased from 31% to 67%, more than double
2. Worker 8 utilization increased from 39% to 73%, nearly double
3. Worker 14 utilization increased from 68% to 77%, slightly more than a 10% improvement
4. Ninety safes of type 1 were produced
5. Fifty safes of type 2 were produced

Notably, all of improvements (1) – (3) brought the utilization in question nearer the traditional 80% which is a good theoretical compromise between low utilization and excessively long queues and wait times therein.

The fourth and final improvement undertaken during this study involved enhancement of the changeover procedure. Whenever a new part type (change from safes of type 1 to safes of type 2 or vice versa) occurred, the workstation must be empty to make required tooling adjustments. Between any two shifts, half an hour is dedicated to these changeovers. The enhancement consisted of having a “deliberately empty workstation” during assembly – that is, “don't load the first (new type) part on the assembly line until the conveyor has indexed once.” Thus, each workstation gets a “breathing spell” cycle slightly shorter than eight minutes, and a changeover requires less than five minutes (providing slack). When the modified schedule was incorporated into the model, the number of type 1 safes remained ninety, and the number of type 2 safes produced rose from fifty to sixty.

VI. CONCLUSIONS AND FURTHER WORK

A sequence of three successive improvements to the process under study, the second and third building upon the previous, significantly improved productivity, inventory levels, and worker utilization percentages. Furthermore, the client is now persuaded of the analytical capabilities and powers of discrete-event process simulation, and hence has already begun to explore its use in additional “continuous improvement” endeavors.

Further explorations are planned, including optimizing the inventory for other components of the safes (e.g., lugs, stiffeners, and joint strips). Also, further financial analyses are planned to optimize the costs of operating the assembly line.

ACKNOWLEDGMENTS

S. Ladge gratefully acknowledges contributions of a fellow researcher who has requested anonymity. Both authors acknowledge the interest, collaboration, and willing cooperation of engineers and managers at the client enterprise where this study was undertaken. Additionally, both authors express their gratitude for the helpful criticisms of several anonymous referees, which have greatly improved the presentation of this work.

REFERENCES

- [1] Law, Averill M. And Michael G. McComas. 1999. Simulation of Manufacturing Systems. In *Proceedings of the 1999 Winter Simulation Conference*, Volume 1, eds. Phillip A. Farrington, Harriet Black Nembhard, David T. Sturrock, and Gerald W. Evans, 56-59.
- [2] Zülch, Gert and Michael Zülch. 2014. Planning Hybrid U-Shaped Assembly Systems Using Heuristics and Simulation. In *Proceedings of the 2014 Winter Simulation Conference*, eds. A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 2180-2191.
- [3] Herrmann, Frank. 2013. Simulation Based Priority Rules for Scheduling of a Flow Shop with Simultaneously Loaded Stations. In *Proceedings of the 27th European Conference on Modelling and Simulation*, eds. Webjørn Rekdalsbakken, Robin T. Bye, and Houxiang Zhang.
- [4] Mosca, Roberto, Filippo Queirolo, and Flavio Tonelli. 2002. Job Sequencing Problem in a Semi-Automated Production Process. In *Proceedings of the 14th European Simulation Symposium*, eds. Alexander Verbraeck and Wilfried Klug, 343-347.
- [5] Gunal, Ali K., Shigeru Sadakane, and Edward J. Williams. 1996. Modeling of Chain Conveyors and their Equipment Interfaces. In *Proceedings of the 1996 Winter Simulation Conference*, eds. John M. Charnes, Douglas J. Morrice, Daniel T. Brunner, and James J. Swain, 1107-1114.
- [6] Kroemer, K. H. E. and E. Grandjean. 1997. *Fitting the Task to the Human: A Textbook of Occupational Ergonomics*. Philadelphia, Pennsylvania: Taylor and Francis Ltd.
- [7] Chung, Christopher A. 2004. *Simulation Modelling Handbook*. Boca Raton, Louisiana: CRC Press.
- [8] Leemis, Lawrence M. 2002. Stat::Fit: Fitting Continuous and Discrete Distributions to Data. In *OR/MS Today* (29,3) [June]
- [9] Thiesing, Renee and C. Dennis Pegden. 2014. Introduction to Simio. In *Proceedings of the 2014 Winter Simulation Conference*, eds. Andreas Tolk, Saikou Y. Diallo, Ilya O. Ryzhov, and Levent Yilmaz, 4192-4201.
- [10] Kelton, W. David, Jeffrey Smith, and David Sturrock. 2013. *Simio and Simulation: Modeling, Analysis, Applications*, 3rd edition. Learning Solutions.
- [11] Hugan, Joseph C. 2014. A Practical Look at Simulation Project Management. In *Proceedings of the 2014 Winter Simulation Conference*, eds. Andreas Tolk, Saikou Y. Diallo, Ilya O. Ryzhov, and Levent Yilmaz, 98-102.
- [12] Groover, Mikell P. 2012. *Fundamentals of Modern Manufacturing: Materials, Processes, and Systems*, 5th edition. New York, New York: John Wiley & Sons, Incorporated.

APPENDIX

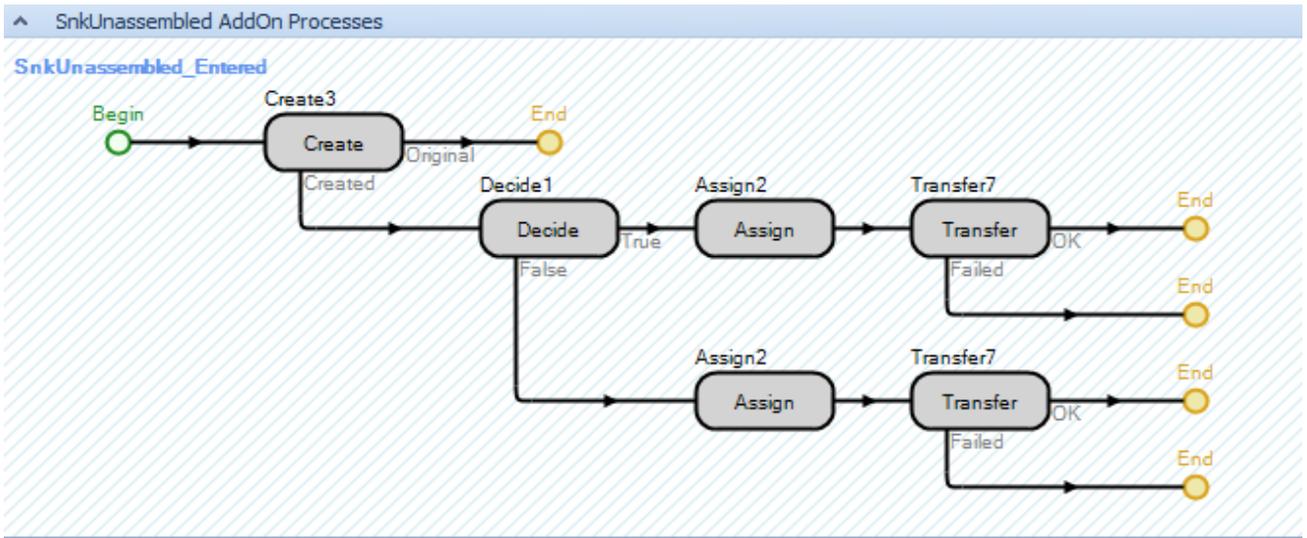


Figure 1. Shift Changeover Logic; Loading New Shell Type at Start of a Shift



Figure 2. Model Animation Showing the Revised “U”-Shape of the Assembly Line

Efficient Modelling and Real-Time Optimisation of Stationary Systems: Application to an Evaporation Process

Carlos Gómez Palacín, José Luis Pitarch and César De Prada

Systems Engineering and Control Department, Escuela de Ingenierías Industriales, Universidad de Valladolid,
Valladolid, Spain.

Email: {carlos.gomez, jose.pitarch, prada}@autom.uva.es

Abstract—Process optimisation usually leads to a nonconvex mathematical problem, which can be solved in an iterative way by using relevant information about the gradients of the objective function and constraints. In this way, the use of automatic-differentiation tools (AD) is recommended to efficiently obtain such information from a system modelled in steady state. This paper deals with the integration problem between some existent powerful modelling & simulation software and tools for numerical optimal control. The approach here is programming a communication interface between EcosimPro® (modelling), CppAD (AD for C++ code) and IPOPT (Interior-Point Optimiser). The effectiveness of the proposed combined tool has been tested in an industrial evaporation process.

Keywords—Stationary model; EcosimPro; CppAD; RTO; NLP; Automatic differentiation; Resource efficiency.

I. INTRODUCTION

The efficient use of resources has become one of the main goals in modern industry [1]. In this framework, optimisation problems naturally arise when dealing with process efficiency. Thereby, real-time optimisation (RTO) of processes (i.e., finding the best operating conditions) is a requirement: it reduces the production cost and the environmental impact. RTO is a wide applied online activity in large-scale systems such as petroleum refineries and chemical plants [2]. As RTO problems are solved at regular intervals, very detailed dynamic models can be replaced by simplified stationary ones. As a result, the optimisation problem is generally large, with many constraints from the process model, but relatively few degrees of freedom for optimisation [3].

In order to run an optimisation problem, some quantitative computations are required to measure the goodness of a process. A very extended one is the so-called key performance indicators (KPI) [4]. Nevertheless, new indicators devoted specially to measure resource efficiency in real time (REI) have been recently developed [5]. They focus in a simpler visualization and aggregation of the information between production aspects and employed energy/materials.

After the definition of such REIs, a large amount of model-based optimisation problems on resource efficiency can be stated and solved with nonlinear programming (NLP) software. These tools usually require precise computation of Jacobians and Hessians to provide the optimiser with accurate information to advance through the right way to the optimum [3][6]. There exist several options to compute derivatives in an automatic way. The most used is the approximation by finite differences, which offers a simple and systematic methodology. Nevertheless, the obtained results are not as accurate

as desired. Another option is symbolic calculus. This option gives exact results at the price of increasing the computational cost, which may become a bottle-neck in applications with demanding real-time constraints. The current trend is to use tools which implement automatic differentiation [7]. Such tools take advantage of the well-known chain rule to automatically codify the required computations for derivatives directly in machine code. Hence, precise results can be obtained in less computational time than the required for the symbolic calculus [8].

Also good object-oriented software tools for modelling and simulation have been developed in recent years [9][10]. These tools already allow to deal with dynamic optimisation problems (differential algebraic equations) by following a sequential approach [11]. However, these tools still fail in solving with pure discrete or steady-state nonlinear models (presence of algebraic loops), as numerical integration algorithms (e.g., [12]) only provide information about sensitivities in systems with differential equations. Automatic-differentiation tools [8][13][14] are therefore required for those cases. Unfortunately, the lack of integration between such tools and the software for modelling/simulation can make the resolution of the overall problem a very tedious task. Indeed, the difference between programming structures and languages employed force the user to redefine almost the whole optimisation problem each time there is a small change on the process and/or the objective function. This issue is addressed in this work in a preliminary way, proposing a two-way communication interface integrating the commercial tool EcosimPro® (System Modeling and Simulation Software) with an NLP optimisation framework implementing AD, as it is CppAD together with IPOPT (optimisation solver for large-scale nonlinear problems).

The rest of the paper is structured as follows: next section explains the programmed communication interface, Section 3 shows the application of the combined tool to an industrial evaporation process, the obtained results after optimising this case study are shown in Section 4 and, finally, a conclusion section closes the paper.

II. INTERFACE DESIGN

This work mainly focuses on the creation of a generic interface for optimisation of stationary systems. This interface will make the optimisation results accessible from external software, like environments devoted to numerical integration. Therefore, the interface will act as a communication system to

provide information from the simulation tool to the NLP optimiser and vice versa. In this way, we will be able to initialise or modify process variables and to recover the values obtained when it is executed. Consequently, the code has to be compiled as a dynamic-link library which can be executed later on inside any simulation software, with the aim of testing performance before implementing the real on-site control algorithms.

By using the inheritance provided by the object-oriented paradigm, we can facilitate the coding of each optimisation problem with a similar interface.

A. Software packages

In this work we decided to use EcosimPro® as modelling & simulation tool, therefore some decisions and developments below are made particularly for this software. Nevertheless, there is no loss of generality due to such choice, because the underlying basic ideas are also valid with other software implementations.

EcosimPro® offers both; [A] a high-level programming language in order to code dynamic models and [B] interfaces to modern numerical integration algorithms. The modelling can be done graphically or with a C++ alike text language. The graphic programming option is only available for certain systems, that have been already defined either by the software development team, or by the user. Also, both styles can be mixed with ease. It can use external functions and classes definitions, with the only need of dynamic-link and static libraries and a header file for the classes definitions. In summary, EcosimPro® is a very powerful tool for modelling/simulation which already incorporates algorithms for dynamic systems optimisation, including the computation of exact gradients via sensitivities. However, derivatives cannot be computed in an exact way for optimisation of stationary models, resulting in a loss of performance. This issue limits also the use of efficient NLP solvers like IPOPT [6], which requires good information about first and second derivatives as input.

In order to incorporate support for efficient optimisation methods linked to stationary models, we suggest here the use of the optimisation tool inside the simulator by using libraries. In order to perform a stationary model-based optimisation, we chose the interior-point NLP solver IPOPT for this work. This package has been proven efficient in large-scale nonlinear optimisations. It is supplied with public license (GPL) under the project Coin-Or, that pretends to give free access mathematical tools globally. Although IPOPT can be ran from several ways, we will use its C++ interface in order to create the files previously defined as needed.

IPOPT is able to reach a local optimum spending less iterations than other gradient-based algorithms thanks to, not only asking for the first derivatives with respect to the decision variables, but also for the second ones. Therefore, it will obtain faster and more accurate results if this information is given with precision.

To efficiently compute and provide derivatives, we chosen the tool CppAD [7], because it implements automatic differentiation in C++ source and it is also maintained as part of Coin-Or project. It provides matrix work functions with derivatives, making the computations of Jacobians and Hessians easier. This software implements AD by using an internal data type to record the operations chain. Then, it

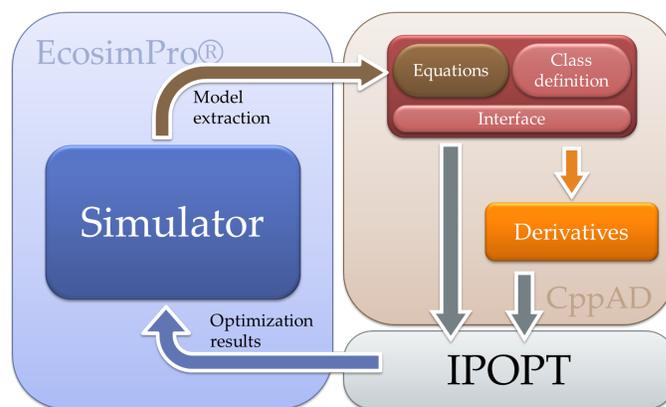


Figure 1. Overall resolution procedure.

can make the differentiation with forward or reverse direction. Forward direction has a calculus time expended around three times the number of variables. While, the reverse option is related to three times the time expended in computing the functions. This means that the appropriate choice will depend on the kind of problem, i.e., for a small problem with a huge number of variables it's better to choose a reverse calculus and for a large problem with very few decision variables is better to work with the forward option. Although CppAD is quite powerful, it doesn't have interfaces to any numerical integration algorithm. Therefore, it cannot be used to solve dynamic problems as it is.

B. Model extraction

First, let the user generate a dynamic model with EcosimPro® in a very intuitive way: equations don't need to hold causality neither a syntax order. Then, once the model is correctly finished, the simulation tool translates it internally to a causal mathematical one. This last model is saved as an *html* file. However, this file can't be directly used by AD tools; it needs a previous translation phase to be coded into the proper programming language (e.g., C++). In this work this translation is done almost completely by a command-line translator, developed to implement the previously described interface from the mathematical-view file. See Figure 1 for a clearer representation of the overall procedure.

Our translator runs over the command line under a 64-bit Windows® OS. It was compiled in this way in order to take advance of the improved memory-handling capabilities of 64 bits systems, however a 32-bit version can be compiled too.

The translator works as follows: first, all the model variables are found and classified as internal variables (the ones that might be modified during the optimisation), data variables (fixed values) or decision variables. Then, the equations are translated into C++ language and, as EcosimPro® already has arranged them by type, the ones that have to be constraints in the stationary model are chosen. The dynamics (if there is any) is removed (by setting the derivatives to zero on the corresponding state equations) to get a stationary model.

The auxiliary functions, which have been already coded using the modelling software (i.e., those belonging to separate libraries in EcosimPro®), have to be translated to CppAD code too. However, their code is not included in the main *html* file,

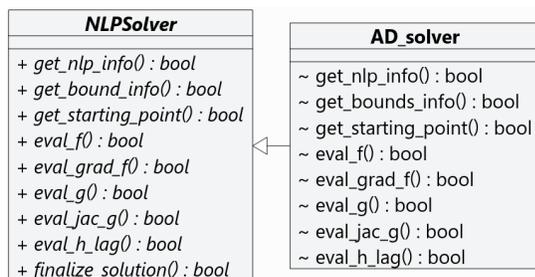


Figure 2. Communication with the optimiser.

despite being used in the model equations. Only the function's name together with its corresponding library is included as a label in the *html* file (e.g., the function *maximum* coded inside the *math* library will be labelled as *math.maximum*). This translation procedure can be done by slightly modifying a plain C++ code that EcosimPro[®] automatically creates each time the functions are defined. However, due to the issue of finding the particular path of these C++ files in each computer, this step is not fully automatised yet.

C. Integrating the optimiser

In order to use IPOPT as a C++ coded program, the methods for an already defined abstract class inside IPOPT have to be implemented. It has to offer a function-call in order to obtain the cost-function and constraints values, besides the Jacobian and the Hessian. All these calculations are implemented using the automatic differentiation algorithm of CppAD, as shown in Figure 2. It will also provide the automation kernel with other valuable information about the problem, as the number of constraints or the limits for those and decision variables. It also has to calculate the sparsity of the Jacobian and Hessians. This is a very important point in order to actually take advantage of all the IPOPT potential.

Therefore, once we have applied the translator to the dynamic model that EcosimPro[®] previously generated, we have to mix it with the solver interface. This class will be different for each optimisation problem, as it may have different targets: the cost function can't be the same to optimise, for instance the required power, than to execute a data-reconciliation procedure. Despite this, the general root class for each model can be the same, modifying only the few different aspects that have been listed before. In this way, we can state different particular optimisation problems, all of them inheriting from the same stationary model. This idea is summarized in the Figure 3.

Then, when the cost function and constraints are introduced, the dynamic-link library has to be compiled with the chosen compiler. In this particular case, we used MSVC++[®].

As soon as we get the compiled code, we can execute it inside external software. Before running the optimisation, in order to obtain reasonable results, the initial guess for decision variables plus upper and lower limits for them and the rest of constraints have to be set up. They can be stated in the same EcosimPro[®] experiment and then read before calling IPOPT. In order to facilitate this communication, all the variables are addressed by strings in CppAD. The constraints can be labelled also with strings, but by default they are indexed by their position. Instead of using C++ index type, where the zero is

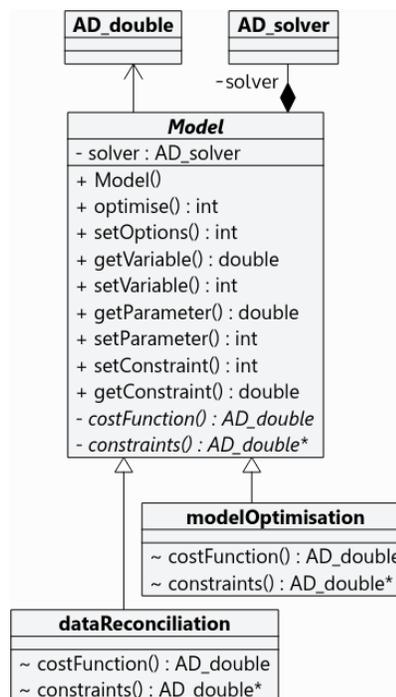


Figure 3. Full classes diagram.

TABLE I. MAIN RETURNED VALUES.

Value	Meaning
1	Successful operation.
2	Maximum iterations achieve.
3	The progression is too small.
4	Enough accurate solution.
5	Infeasible problem.
12	Not enough freedom degrees.

the first, we chosen one as the first to keep the EcosimPro[®] index style.

Henceforth, with all the initialisation done, the optimisation can be executed and runs in a totally transparent way for the end user. This procedure will return an integer depending on the result of the internal computation. The more common outputs are listed in Table I. If the problem finds a local optimum, we can recover the decision-variables values in order to use them in simulation to check if it approaches the prediction given by the optimiser. In this way, we can pretend the behaviour on a real plant by using the simulation variables as sensor signals, changing model parameters or external input disturbances and updating the control inputs with the decision-variables values.

III. INDUSTRIAL APPLICATION

The software developments presented in the above section have been applied in practice to optimise the hot steam consumption in a multiple-effect evaporation process. This process is formed by: two sets of evaporation chambers and two sets of heat exchangers in serial connection, a barometric condenser, a cooling tower and a saturator (see Figure 4). The system receives and recirculates a liquid mixture of water plus

several chemical components with the aim of concentrating it by evaporating a certain amount of water (to partially recover such chemicals by later crystallization).

A. Mathematical model

A grey-box model has been created to represent the process in steady state. The main part is based on first principles:

- Energy and mass balances in the main equipment (evaporators, heat exchangers and condenser):

$$\sum_{i=1}^{n_i} F_i = \sum_{j=1}^{n_o} F_j \quad (1)$$

$$\sum_{i=1}^{n_i} F_i \times H(T_i, C_i) - \sum_{j=1}^{n_o} F_j \times H(T_j, C_j) = K_p \times [\bar{T} - T_{amb}] \quad (2)$$

Where F_i , F_j are the n_i input or n_o output mass flows from each equipment respectively, $H(T, C)$ is the specific enthalpy function, T_{amb} is the ambient temperature, \bar{T} can be the average or maximum temperature in each equipment and K_p is a coefficient representing the heat loss to ambient.

- Density relationships between mass and volumetric flows of the mixture, water and steam as a function of temperature and/or pressure (omitted for brevity).
- Heat transmission between fluids inside the heat exchangers:

$$Q = UA \times LMTD \quad (3)$$

$$LMTD = \left[\Delta T_1 \times \Delta T_2 \frac{\Delta T_1 + \Delta T_2}{2} \right]^{\frac{1}{3}} \quad (4)$$

Where Q is the heat power, U is the heat transmission coefficient and the *mean-logarithmic temperature difference* ΔT has been computed using the Chen's approximation [15].

- Chemical equilibrium relationships between pressure-temperature-concentration of the mixture inside the evaporation chambers:

$$\bar{P} = P_v(\bar{T}_{mix}, \bar{C}), \quad \bar{T}_{steam} = T_v(\bar{P}) \quad (5)$$

Where $P_v(T, C)$ and $T_v(P)$ are functions to compute the vapour pressure of the mixture and the saturated steam temperature respectively, and \bar{T}_{mix} , \bar{C} , \bar{T}_{steam} , \bar{P} are averaged temperature and concentration of the mixture and averaged temperature and pressure of the saturated steam respectively, all inside a chamber.

Note that these temperatures and pressures to be used here do not correspond to particular variables in the real plant, i.e., they are fictitious variables corresponding to a mixture of real *unknown* ones.

In addition, and due to the fact that actual processes are of high order and knowing all relationships between internal variables may be difficult, the above physical laws are complemented with experimental patterns. Data reconciliation techniques [16] have been used to identify model unknown parameters, such as the heat-transmission coefficients, and

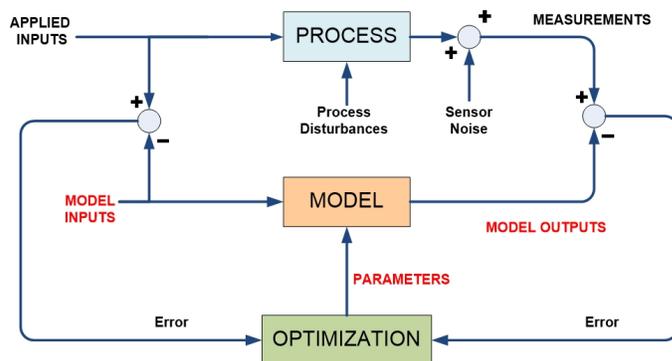


Figure 5. Schema of the data reconciliation procedure.

possible patterns/relationships between internal variables (non-measurable) from a set of measurements with the following procedure:

- 1) Data pretreatment of raw measurements in order to exclude outliers (out-of-range values or sensor failure).
- 2) Parameter and process variables estimation by solving the data reconciliation problem with the model based on first principles.
- 3) Experimental identification of patterns.
- 4) Complete model (physical and experimental laws) validation by data reconciliation with a different set of measurements.

With the aim of reducing the effect of an eventual presence of gross errors in the measurements, robust estimators, like the so-called *fair function* [17], have been used as objective function for the data reconciliation instead of the standard least-squares one.

Consequently with the above procedure, set of tests have been done to collect data from the evaporator line in continuous operation. The tests were executed by running the evaporator either:

- each week choosing different pairings for the set points of the control variables and waiting for the steady state,
- at a constant level between two cleaning cycles.

Note importantly that each variable in the model (input, output, state or parameter) is actually considered as decision variable for the data-reconciliation setup (see Figure 5).

At the end stage, the proposed nonlinear stationary model is formed by 46 equations (6 algebraic loops appearing) and has been successfully validated with plant measurements recorded during 8 months of normal operation.

B. Optimising the efficiency

The goal here is moving the process to an operation point where the resource consumption (mainly the hot steam flow) is minimised by the selection of optimal control set points. To achieve this, 3 REIs have been defined for this process:

Definition 1. Denote by *specific steam consumption* (REI_1) to the overheated steam consumed by amount of evaporated water. Then, the *relative steam consumption* will denote the

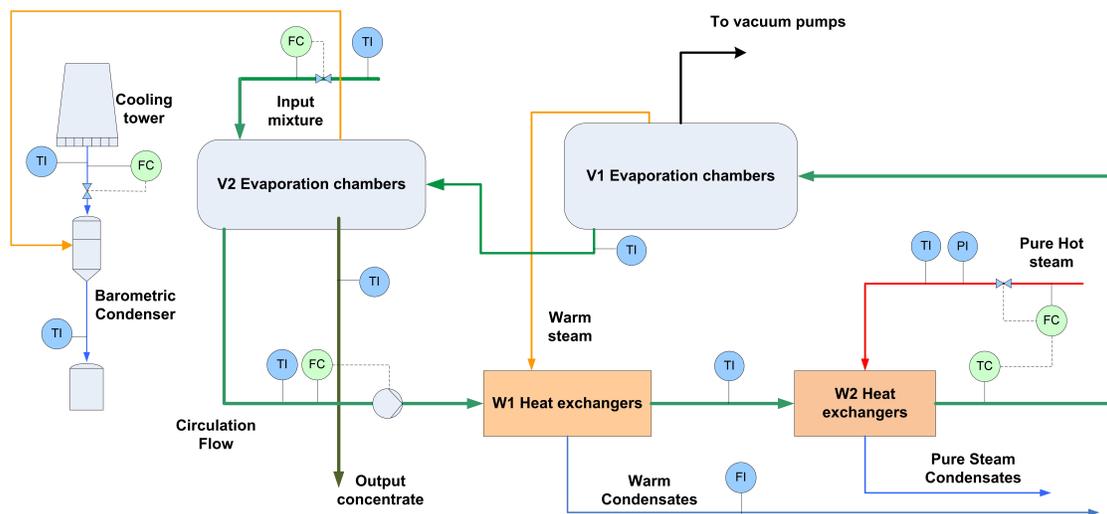


Figure 4. Schema of the evaporation process with existent instrumentation.

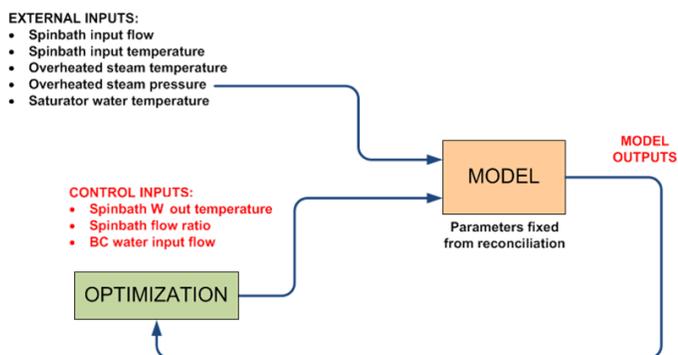


Figure 6. Stated RTO problem.

actual specific steam consumption compared with an historical reference case (REI_2), or with the result of a model-based optimisation (REI_3).

$$REI_2 = \frac{Actual\ REI_1}{Hist\ REI_1} 100 \quad REI_3 = \frac{Actual\ REI_1}{Opt\ REI_1} 100$$

The following RTO problem [3] has been stated to compute the REI_3 . Minimise $J(u)$ subject to:

- Evaporate a desired rate of water
- The model equations
- Saturation in the control actions
- A limit on the available cooling power

Where the objective function $J(u)$ to minimise is the overheated steam flow and the decision variables u are the optimal set points to the controllers (see Figure 6).

The implementation of this modelling & optimisation problem is carried out by using the software tools & developments presented in Section 2. Then, the idea is to periodically perform an automatic RTO with adequate frequency, e.g., one reasonable enough to update the control actions.

IV. RESULTS

With the aim of analysing the results given by the optimisation setup stated in the above section, a set of experimental data has been recorded with the process in steady state in different operation points. The optimiser has been fed with the reconciled values of the process inputs (flow and temperature of the input mixture, ambient temperature, pressure and temperature of the overheated steam input, etc.), solving a RTO problem for each one of the sample instants taken.

The computed optimal values for the control set points show a clear trend: the higher the temperature and the lower the circulating flow are, the better the process efficiency is. Figure 7 shows a simulation of the dynamic response of the process (obtained with a more detailed dynamic model, different from the simplified stationary one for optimisation) with a change of the current control set points to the optimal ones predicted by the optimiser. The visualisation of REI_1 for different sample points out that average savings around a 3% might be obtained (see Figure 8) with the optimal operation (controller set points).

Be aware that the optimisation setup stated by means of the proposed automatic-translation tool, reaches the same solution than the one obtained by a manual programming approach.

Note that actual experimental data is not included due to a confidentiality agreement with the industrial partner. Values

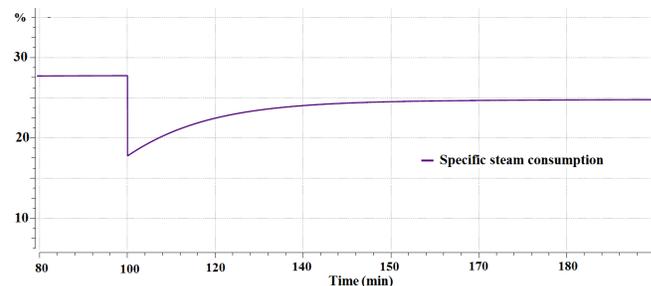


Figure 7. Evolution of the steam consumption before and after applying the optimal control set points.

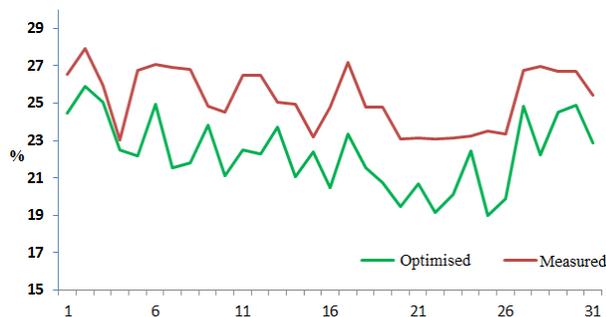


Figure 8. Evolution of the specific steam consumption with sampling.

in Figures 7 and 8 are also scaled in % by the same reason.

Note on computational cost. The resulting NLP problem for this case study has 21 decision variables and 39 constraints. It is solved in less than 0.1s, running in an Intel® Core™ i3-2310M under Windows 7® 64 bits. Therefore, as the transient duration of the process is 45 minutes approximately, this RTO problem can be periodically launched (e.g., each hour) in any standard machine without compromising real-time constraints.

V. CONCLUSION

The high potential offered by the current software for modelling and simulation makes the use of it desirable in other advanced tasks, such as optimisation. In this sense, modern NLP tools for numerical optimal control allow obtaining efficient solutions for optimisation of actual industrial processes.

This work proposed a communication interface between both specialised software tools in order to facilitate the model translation into an optimisation problem and also to test the proposed *theoretically optimal* solutions later in simulation. In comparison with a manual procedure, this automatic translation tool saves time in the initial model codification and its further possible modifications. In addition, it avoids translation errors and loss of data associated to the human factor. Furthermore, the proposed interface, together with the powerful existent software, opens the door to other demanding applications in which solving large-scale problems in short periods of time is required.

The developments in this work have been successfully applied to address a problem of resource efficiency in an evaporation process, very common in current industry. However, the obtained predictions of savings are, obviously, overly optimistic (existence of mismodelling, etc.). Therefore, they only give qualitative information about a possible right way to improve. The proposed recommendations for the control actions must be implemented in the control system to be evaluated on site.

Obviously, one case study is not enough to completely validate the proposed interface, as there may be complicated process models leading to unexpected situations in the translation procedure. Nevertheless, this issue is very common in software development, where further feedback and polishing stages are needed after a product launch, and it does not detract the usefulness of the approach.

Further work will be focused on increasing robustness to mismodelling and external disturbances by implementing stochastic optimisations and modifier-adaptation setups.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n°604068 and from the Spanish MINECO under project DPI2012-37859.

REFERENCES

- [1] J. Dahmus, "Can efficiency improvements reduce resource consumption?" *Journal of Industrial Ecology*, vol. 18, no. 6, 2014, pp. 883–897, ISSN: 1530-9290.
- [2] J. Han, G. S. Forman, A. Elgowainy, H. Cai, M. Wang, and V. B. DiVita, "A comparative assessment of resource efficiency in petroleum refining," *Fuel*, vol. 157, 2015, pp. 292 – 298, ISSN: 0016-2361.
- [3] L. Biegler, *Nonlinear Programming*. Society for Industrial and Applied Mathematics, 2010, ISBN: 0898717027, 9780898717020.
- [4] D. Parmenter, *Key performance indicators (KPI): developing, implementing, and using winning KPIs*. John Wiley & Sons, 2010, ISBN: 9780470095881.
- [5] M. Kalliski, D. Krahé, B. Beisheim, S. Krämer, and S. Engell, "Resource efficiency indicators for real-time monitoring and optimization of integrated chemical production plants," in *25th Euro. Sym. on Computer Aided Process Eng.*, ser. *Computer Aided Chemical Engineering*, K. V. Germaey, J. K. Huusom, and R. Gani, Eds. Elsevier, 2015, vol. 37, pp. 1949 – 1954, ISSN: 1570-7946.
- [6] A. Wächter and L. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Programming*, vol. 106, no. 1, 2006, pp. 25–57, ISSN: 0025-5610.
- [7] B. Bell and J. Burke, "Algorithmic differentiation of implicit functions and optimal values," in *Advances in Automatic Differentiation*, ser. *Lecture Notes in Computational Science and Engineering*, C. Bischof, H. Becker, P. Hovland, U. Naumann, and J. Utke, Eds. Springer Berlin Heidelberg, 2008, vol. 64, pp. 67–77, ISBN: 978-3-540-68935-5.
- [8] X. Li, Z. Shao, and J. Qian, "Module-oriented automatic differentiation in chemical process systems optimization," *Computers & Chemical Engineering*, vol. 28, no. 9, 2004, pp. 1551 – 1561, ISSN: 0098-1354.
- [9] "EcosimPro. Dynamic Modeling & Simulation Tool," URL: <http://www.ecosimpro.com/> [accessed: 2015-09-09].
- [10] "Modelica. A Unified Object-Oriented Language for Physical Systems Modeling," developed by the Modelica Association. URL: <https://www.modelica.org/> [accessed: 2015-09-09].
- [11] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, vol. 47, no. 1, Jan. 2005, pp. 99–131, ISSN: 0036-1445, 1095-7200.
- [12] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward, "Sundials: Suite of nonlinear and differential/algebraic equation solvers," *ACM Trans. Math. Softw.*, vol. 31, no. 3, Sep. 2005, pp. 363–396, ISSN: 0098-3500.
- [13] J. Andersson, J. Åkesson, and M. Diehl, "Casadi: A symbolic package for automatic differentiation and optimal control," in *Recent Advances in Algorithmic Differentiation*, ser. *Lecture Notes in Comp. Science and Engineering*, S. Forth, P. Hovland, E. Phipps, J. Utke, and A. Walther, Eds. Springer Berlin Heidelberg, 2012, vol. 87, pp. 297–307, ISBN: 978-3-642-30022-6.
- [14] A. Walther and A. Griewank, "Getting started with ADOL-C," in *Combinatorial Scientific Computing*, U. Naumann and O. Schenk, Eds. Chapman-Hall CRC Computational Science, 2012, pp. 181–202, ISBN: 978-1-4398-2735-2.
- [15] J. J. Chen, "Comments on improvements on a replacement for the logarithmic mean," *Chemical Engineering Science*, vol. 42, no. 10, 1987, pp. 2488–2489, ISSN: 0009-2509.
- [16] M. Leibman, T. Edgar, and L. Lasdon, "Efficient data reconciliation and estimation for dynamic processes using nonlinear programming techniques," *Computers & Chemical Engineering*, vol. 16, no. 1011, 1992, pp. 963 – 986, ISSN: 0098-1354.
- [17] P. Huber, "Robust statistics," in *International Encyclopedia of Statistical Science*, M. Lovric, Ed. Springer Berlin Heidelberg, 2014, pp. 1248–1251, ISBN: 978-3-642-04897-5.

An Analytical Model to Evaluate the Response Capacity of Emergency Departments in Extreme Situations

Work-in-Progress Paper

Eva Bruballa, Manel Taboada

Tomàs Cerdà Computer Science School
Universitat Autònoma de Barcelona (UAB)
Barcelona, Spain

e-mail: eva.bruballa@eug.es, manel.taboada@eug.es

Alvaro Wong, Dolores Rexachs, Emilio Luque

Computer Architecture and Operating Systems Department
Universitat Autònoma de Barcelona (UAB)
Barcelona, Spain

e-mail: alvaro@caos.uab.es, dolores.rexachs@uab.es,
emilio.luque@uab.es

Abstract—One of the most important current problems in the healthcare system is the saturation of Emergency Departments, due to the increasing demand of the service. Healthcare staff often has no margin to absorb the demand in case of an unexpected increase of patients entering the service caused by an emergency. We aim to evaluate the response capacity of an Emergency Department, specifically of doctors, nurses and specialist technicians that make up a specific sanitary staff configuration, facing an unexpected emergency. Such situations cannot be tested in the real system and simulation is the only way to obtain data about them. We present an analytical model to obtain information from data obtained through the simulation of a Hospital Emergency Department. The model defines how to calculate the theoretical throughput of a particular sanitary staff configuration, that is, the number of patients it can attend per unit time given its composition. This index is a reference in order to measure the emergency response capacity of the system and also other indicators concerning to performance. The data for the analysis will be generated by the simulation of any possible scenario of the real system, taking into account all valid sanitary staff configurations and different number of patients entering into the emergency service. The designed model offers the availability of relevant knowledge to the managers of the service to make decisions concerning the composition of the sanitary staff configuration in anticipation of extraordinary situations.

Keywords—Agent-Based Modeling and Simulation (ABMS); Decision Support Systems (DSS); Emergency Response Capacity; Emergency Department (ED); Knowledge Discovery.

I. INTRODUCTION

Currently, given the growing demand of emergency medical care, the management of Emergency Departments (ED) is increasingly important. Particularly, how to manage the increasing number of patients entering into the service is one of the most important problems in the ED worldwide, because it requires a substantial amount of human and material resources, which unfortunately are often too limited, as well as a high degree of coordination between them [1][2]. A major consequence of the increase of patients entering the service is its saturation [3]. This results in an increase in the

length of stay of patients in the service, which produces a general discontent among them, abandoning patients without receiving care, limited access to emergency care and an increasing patient mortality [4].

A sudden or unexpected increase in the entrance of patients is an extraordinary, but possible situation, which most likely cause the collapse of the service. The response capacity of the system in anticipation of such situations is the leeway of the sanitary staff configuration to absorb an extra demand of the service, and it will depend on its composition. Also, the waiting time of patients for medical attention and the total time spent in the service, which are indicators of the quality of service from users' point of view, will heavily depend on the configuration composition.

Even though computerized systems in hospitals provide more and more real data, there are no real data available to analyze concerning this kind of situations, which cannot be tested in the real system, but this limitation can be overcome through computer simulation.

There are many examples of computer simulation models that support decision-making processes in the health sector [5][6][7][8]. Specifically, simulation has been widely used to understand causes of ED overcrowding and to test interventions to alleviate its effects, and many contributions show how simulation can be used as a tool to address this problem. An extensive review of this related work is done in [9].

Our contribution in this context consists on using simulation as a sensor of the real system, being our main source of data, an ED simulator, based on an Agent-Based Modeling (ABM) design of the system [10]. The simulator has been developed, verified and validated by the "High Performance Computing for Efficient Applications and Simulation" Research Group (HPC4EAS) of the Universitat Autònoma de Barcelona (UAB) in collaboration with the Hospital de Sabadell (Parc Taulí, Barcelona). The input parameters that characterize each different scenario in the simulation of the real system are the healthcare staff configuration, the number of incoming patients, the derivation percentage of patients and the period of time simulated. In addition, each simulation provides data of the length of stay of all patients in all locations in the ED and the

number of patients per hour and location [11][12]. The number and variety of simulated agents and the different possible values for the input parameters of the simulator, give a great number of different possible scenarios to execute. Each execution will generate the corresponding data based on the designed model.

The paper is organized as follows: Section II presents the research objectives and methodology; Section III describes the Emergency Department process; Section IV presents the analytical model proposed. Finally, Section V closes the paper with discussion and future work.

II. RESEARCH OBJECTIVES AND METHODOLOGY

The general purpose of our research is to gain knowledge about the variables that may influence in the performance and the quality of service of an emergency department from the data generated via simulation of the real system for any possible situation.

Specifically, the goal of this work is to develop an analytical model to obtain information about the emergency response capacity of the system. This will provide staff demand prediction models in the service, especially for unusual situations, as a reference to make decisions and to anticipate solutions in exceptional cases.

The simulator will be the main source of data. These data are the raw material for the analysis. When the user assigns them some special meaning, they become information. We refer to as knowledge when a model is found or designed, in order to interpret this information, and the model represents an added value [13]. Therefore, the analysis of the data generated by the simulator can give us prediction and behavior models for an emergency department for any possible situation. In particular, our specific objective is to determine the features associated to a sanitary staff configuration to ensure an acceptable response capacity in anticipation of a possible emergency situation.

With this objective in mind, an analytical model has been designed for the processing and analysis of the raw data obtained directly from the simulator.

III. DESCRIPTION OF THE SIMULATION MODEL

A. Emergency Department Process

The operation of the ED is based on a process consisting of different steps or phases on which each patient is passing from its entry into the service until it is discharged, referred to another service or admitted to the hospital.

The ED is divided into different areas, which correspond with the different process phases:

- *Admissions Area:* Administrative staff carries out the registration of the patient's arrival and the reasons for his visit to the emergency service.
- *Triage Area:* Professional sanitary staff identifies the priority level with which the patient should be treated.
- *Diagnosis-Treatment Zone:* Sanitary staff (doctors, nurses and specialist technicians) tries to identify the causes of the patient's health problem, and to the extent possible, try to solve it. This area is in turn

divided into different areas or zones (medical room, nursing room, care boxes and X-ray laboratories).

- *Waiting Rooms:* Distributed in different zones of the ED, where patients wait to be treated at the different stages of the process.

B. Functionality of the Simulator

The simulator includes the following agents: patients, admissions staff, triage nurses, assistant nurses, doctors and radiology technicians. In the case of agents representing sanitary staff (all except patients), we consider two levels of experience (Junior/Senior) and all of them can work in parallel on each phase.

The actions and interactions between the involved agents at each process step result in changes of state of the agents, which ultimately result in the global operation of the system.

From the moment when the patient enters the service, the simulation runs according to the patient flow shown in Figure 1.

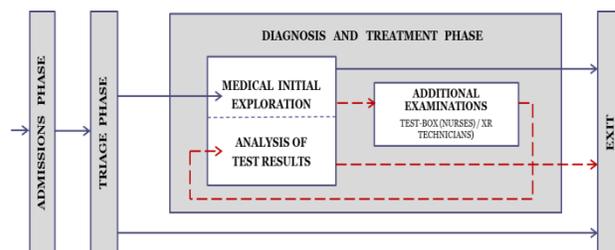


Figure 1. Patient flow in the emergency department.

The admission and triage phases are common to all patients entering the service, and there is a percentage, although low, of patients being referred to other services after triage stage.

In the diagnostic and treatment phase, all patients generated by the system go through an initial medical exploration phase. Some of them are discharged and leave the ED (in blue in Figure 1). The rest remain in the ED and they go through a phase of complementary examinations carried out by technical staff and/or clinical nurses. After this they return again with the doctor who analyzes the test results and, finally, they are discharged from the service (in red and dashed line in Figure 1).

For a given number of patients in the service, the simulation runs for all possible sanitary staff configurations. Each scenario is identified by a sanitary staff configuration and a specific input of patients into the service, and the output of the simulation brings data concerning the number of attended patients, attention time and waiting time for each patient in all phases in their way through the service.

IV. ANALYTICAL MODEL

In order to transform this raw data directly generated by the simulation and to visualize the potential information which they contain, we have defined a set of indexes that characterize numerically each scenario and, in particular, each configuration. These defined indexes facilitate the treatment and analysis of latent information in data.

A. Definition of indexes

As an indicator of the quality of service from the point of view of the user, we define an index called *Patient attention Time (PaT)* as the total time a patient is receiving attention throughout all stages in the service for a given configuration. Another indicator of quality is the *Length of Stay (LoS)* defined as the total length of stay of a patient in the service. And finally the *Length of Waiting (LoW)* is the total waiting time of a patient along the service. Note that,

$$\text{LoS} - \text{PaT} = \text{LoW} \text{ and always } \text{PaT} \leq \text{LoS}. \quad (1)$$

Moreover, the *Equivalent Patient attention Time* for the stage i ($E\text{PaT}_i$) is defined as the attention time of a patient taking into account the possibility of working in parallel for the agents in that stage, and (2) shows how is calculated:

$$E\text{PaT}_i = \frac{1}{\sum_{j=1}^{\text{staff}} \frac{1}{\text{PaT}_{\text{staff}_j}}} \quad (2)$$

where $\text{PaT}_{\text{staff}_j}$ represents the value of the attention time corresponding to each type of sanitary staff, depending also on their level of experience. These values result from the simulator's calibration with the historical real data provided by the hospital.

The slowest phase of the configuration will fix the speed at which patients can be attended in the service and also is the one which can saturate the system. It is, therefore, the equivalent attention time of the slowest phase, which will determine the number of patients that a given configuration can treat per unit of time, given its composition. We call this index *Theoretical Throughput (T_ThP)*, which is an indicator patients' attention capacity of the configuration. Expression (3) gives its calculation:

$$T_ThP = \frac{1}{\text{Max } E\text{PaT}_i} \quad (3)$$

It is also defined the *Real Throughput (R_ThP)* index as the real number of patients treated per unit time. This is an index which values are direct output data from simulation. Then, the ratio between real and theoretical throughput is defined as an indicator of *Performance (PeR)* of a given configuration for a specific scenario:

$$\text{PeR} = \frac{R_ThP}{T_ThP} \quad (4)$$

B. Theoretical throughput for the diagnosis and treatment phase.

Unlike other stages of the process of a patient along his pass through the emergency department, this is the most complex stage due to its nonlinearity. The return of patients for the doctor's final diagnosis after completing complementary examinations requested by the doctor after his first contact with the patient (initial exploration phase), must be taken into account.

We have defined the *Theoretical Throughput (T_ThP)* as an indicator of the attention capacity of each phase, that is, as the number of patients that, for its composition, this phase could treat.

In this phase, the attention time of each type of doctor depending on their experience (Junior or Senior) and depending on the type of care they are doing, either in the first step of initial exploration, or in the second, consisting of the analyzing of the results of a requested supplementary examination, are known and denoted by:

- $\text{IPaT}_{S/J}$: Attention time of a senior/junior doctor in the Initial Exploration phase (I).
- $\text{ARPaT}_{S/J}$: Attention time of a senior/junior doctor in the Analysis of Results phase (AR).

Given these times, their inverse will give us the number of patients that each doctor can treat per unit time considered.

Moreover, we represent with D (*Doctors*) the total number of doctors in the considered combination. Then:

- DSI = Number of senior doctors attending patients in the Initial Exploration phase.
- DSA = Number of senior doctors attending patients in the Analysis of Results phase.
- DJI = Number of junior doctors attending patients in the Initial Exploration phase.
- DJA = Number of junior doctors attending patients in the Analysis of Results phase.

We consider a sufficient rate of patients entering into the service to ensure that the system is running continuously and we assume the system is in a steady state. We also contemplate that doctors prioritize the attention of patients who have already gone through the first exploration phase, and therefore, these patients will be treated at the time the doctor has availability. This prevents endless queues at the return of patients from their requested complementary examination.

Given all these preliminary considerations, in the specific case with only senior doctors, and a percentage p_t of patients who require additional tests, we obtain the following relation of continuity:

$$\frac{\text{DSI}}{\text{IPaT}_S} \times \frac{p_t}{100} = \frac{\text{DSA}}{\text{ARPaT}_S} \quad (7)$$

being $\text{DSI} + \text{DSA} = D$, the total of doctors in the configuration.

Then, the *Theoretical Throughput* of the doctor's combination is calculated in (8):

$$T_ThP_{\text{diag\&trat}} = \frac{\text{DSI}}{\text{IPaT}_S} \times \frac{100-p_t}{100} + \frac{\text{DSA}}{\text{ARPaT}_S} = \frac{\text{DSI}}{\text{IPaT}_S} \quad (8)$$

From (7) and considering $\text{DSI} + \text{DSA} = D$ we can obtain DSI , and then obtain the value for the Theoretical throughput of the diagnosis and treatment phase (diag&trat) in (8), in the considered case.

It rests to make a generalization of the model for any possible combination of doctors (senior or junior).

C. Emergency Response Capacity of the system

We define the Emergency Response Capacity of the system (ERC) as the capacity of a given configuration to provide extraordinary service to patients in an exceptional situation related with an unexpected increase in the number of patients requiring urgent attention.

The desired leeway of the sanitary staff configuration to absorb an extra demand of the service, can be fix in terms of the percentage of patients from its theoretical capacity, that the current configuration should be able to absorb, in expectation of possible but unexpected situations related to a surge in patients entering the service. Considering this, the *Emergency Response Capacity (ERC)* for a given configuration, can be defined as:

$$ERC = T_ThP \times \frac{p_e}{100} \quad (9)$$

where T_ThP is the theoretical capacity (number of patients per unit time) of a specific sanitary staff configuration, or a specific phase, and it has been defined in (3).

This percentage also establishes a maximum throughput for the considered configuration, or specifically, for each stage, in normal situation, that is:

$$ThP_{max} = T_ThP \times \frac{100-p_e}{100} \quad (10)$$

The ultimate goal is to anticipate the best mix between each type of sanitary staff (senior/junior), and the total number of sanitary people to be able to ensure the established emergency response capacity.

V. CONCLUSION AND FUTURE WORK

The initial contribution of the research is the idea of using the simulator of the ED as source of intensive data (sensor of the real system), as it brings the possibility of obtaining data from simulation of any kind of situations, including those which cannot be tested in the real system.

Other contribution to knowledge of the presented work, is the definition of a model to analyze these intensive data to obtain the potential information contained in data, related with the influence of the staff configuration on the efficiency, the performance and the exceptional response capacity of the system.

As immediate future work, in order to check that the proposed model works well, we will try to extract information from a set of data already available, which finally would allow us to obtain relevant knowledge through the values of the defined indexes for all the simulated scenarios.

To do this we will begin working with the data generated by the execution of the simulation of 93780 different scenarios with a reduced version of the simulator [14]: 23445 different staff configurations and four different inputs for the number of patients, one corresponding to a regular situation, and the rest simulating other extraordinary situations. The generated data processing will consist in obtaining the values

of the defined indexes in the model, for all the executed scenarios.

The information from the indexes will allow us to see similarities between configurations with lower values of LoS, which will give us a first approach to the desirable features for a configuration regarding the quality of service from the point of view of the patient, user of the service. To perform this study we plan to use data mining, in particular k-means clustering method, which should allow us to visualize common characteristics between configurations in the same cluster.

Moreover, the analysis of the emergency response capacity and performance of the system, should lead us to identify the characteristics of the most suitable configurations facing an increase in the number of input patients, and therefore, this is desirable knowledge from the point of view of the service manager. It could also be interesting, for further work, the selection of new useful indexes and generation of their models.

Finally, we aim to translate and extrapolate the model to a general model, as the basis for the preprocessing and analysis of data generated by the simulator in its extended version [15].

ACKNOWLEDGMENT

This research has been supported by the MINECO (MICINN) Spain under contracts TIN2011-24384 and TIN2014-53172-P.

REFERENCES

- [1] F. Kadri, S. Chaabane, T. Berger, D. Trentesaux, C. Tahom, and Y. Sallez, "Modelling and management of the strain situations in hospital systems using ORCA approach", IEEE IESM, Rabat, Morocco, October 2013, pp. 1-9.
- [2] F. Kadri, S. Chaabane and C. Tahon, "A simulation-based decision support system to prevent and predict strain situations in emergency department systems", Simulation Modelling Practice and Theory, vol. 42, March 2014, pp. 32-52.
- [3] A. Boyle, K. Beniuk, I. Higginson and P. Atkinson, "Emergency Department Crowding: Time for interventions and policy evaluations", Emergency Medicine International, 2012, pp. 1-8 .
- [4] P.C. Sprivilis, J.A. Da Silva, I.G. Jacobs, A.R.L. Frazer and G.A. Jelinek, "The association between hospital overcrowding and mortality among patients admitted via Western Australian emergency departments", Medical Journal of Australia, vol. 184, num. 5, 2006, pp 208-212.
- [5] B. Mielczarek and J. Uziako-Mydlikowska, "Application of computer simulation modeling in the health care sector: a survey", Simulation, vol. 88, num. 2, 2010, pp. 197-216.
- [6] M.D. Rossetti, G.F. Trzcinski and S.A. Syverud, "Emergency department simulation and determination of optimal attending physician staffing schedules", In: Proceedings of the 1999 Winter Simulation Conference, 1999, pp 1532-1540.
- [7] F.F. Baesler, H.E. Jahnsen and M. DaCosta, "The use of simulation and design of experiments for estimating maximum capacity in an emergency room", In: Proceedings of the 2003 Winter Simulation Conference, 2003, pp 1903-1906.
- [8] S. Samaha, W.S. Armel and D. W. Starks, "The use of simulation to reduce the length of stay in an emergency

- department”, In: Proceedings of the 2003 Winter Simulation Conference, 2003, pp 1907-1911.
- [9] A.P. Sharoda, M.C. Reddy and C.J. DeFlicht, “ A systematic review of simulation studies investigating emergency department overcrowding”, *Simulation*, vol. 86, num. 8-9, 2010, pp. 559-571.
- [10] C.M. Macal and M.J. North, “Introductory tutorial: Agent-based modeling and simulation”, In: Proceedings of the 2013 Winter Simulation Conference, 2013, pp 362-376.
- [11] M. Taboada, E. Cabrera, M. L. Iglesias, F. Epelde and E. Luque, “An agent-based decision support system for hospitals emergency departments”, *Procedia Computer Science*, vol. 4, ICCS 2011, pp. 1870–1879.
- [12] M. Taboada, E. Cabrera and E. Luque, “Modeling, simulation and optimization of resources management in hospital emergency departments using the agent-based approach”, *Advances in Computational Modeling Research*, 2013, pp.1-31.
- [13] L.C. Molina, “Data mining: Torturando a los datos hasta que confiesen”, Fundación Oberta de Catalunya (FUOC), Barcelona, Spain, 2002. Retrieved: september, 2015: http://www.giaa.inf.uc3m.es/miembros/lmarti/_media/master-intel%3Bmolinal102.pdf
- [14] E. Cabrera, M. Taboada, M. L. Iglesias, F. Epelde and E. Luque, “Simulation optimization for healthcare emergency departments”, *Procedia Computer Science*, vol. 9, ICCS 2012, pp. 1464–1473.
- [15] L. Zhengchun, E. Cabrera, D. Rexachs and E. Luque, “A generalized agent-based model to simulate emergency departments”, *The Sixth International Conference on Advances in System Simulations, IARIA, Nice, France, October 2014*, pp. 65-70.

Differentiated Service Queuing Disciplines in NS-3

Robert Chang, Mahdi Rahimi, and Vahab Pournaghshband

Advanced Network and Security Research Laboratory
 California State University, Northridge
 Northridge, California, USA
 {robert.i.chang, mahdi.r.rahimi}@ieee.org
 vahab@csun.edu

Abstract—Network Simulator 3 (ns-3) is a powerful tool for modeling the behavior of computer networks using simulation. We have developed three well known differentiated service packet queuing methods: strict priority queuing, weighted fair queuing, and weighted round robin queuing, in the simulation framework. In this paper, we present the implementation details of the three modules as well as their usage. By implementing these modules in ns-3 and demonstrating their use, we intend to facilitate further research and experimentation with our contributions. We believe that our work will be utilized in solving outstanding problems that would have been impractical to investigate without our modules. Lastly, through validation, we confirm that our introduced modules simulate these queuing methods correctly.

Keywords—strict priority queuing; weighted fair queuing; weighted round robin; differentiated service; ns-3

I. INTRODUCTION

We are presenting three new modules for three scheduling strategies: strict priority queuing, weighted fair queuing, and weighted round robin. These queuing methods offer differentiated service to network traffic flows, optimizing performance based on administrative configurations.

The network simulator 3 (ns-3) [1] is a popular and valuable research tool which can be used to simulate systems and evaluate network protocols. ns-3 organizes components logically into modules. The official modules included by default are able to create basic simulated networks using common protocols, and users can add additional components by creating specialized modules. This has been used to add a leaky bucket scheduler [2] and to add and evaluate a DiffServ framework implementation [3].

DiffServ is a network architecture that provides a way to differentiate and manage network flows. A DiffServ network can give priority to real-time applications, such as Voice over IP, to ensure acceptable performance, or prevent malfunctioning and malicious applications from occupying all of the bandwidth and starving other communication. Two of the main components of DiffServ are classification and scheduling. DiffServ networks classify the packets in a network flow to determine what kind of priority or service to provide and schedule packets according to their classification. Differentiated service queuing disciplines, such as those described in this paper, are responsible for executing the flow controls required by DiffServ networks.

This paper is organized as follows: first, a brief overview of the theoretical background behind each of our modules is

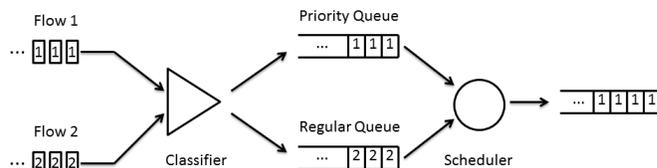


Figure 1. A strict priority queue

presented in Section II. In Section III, we overview existing simulation tools for differentiated service queuing. Section III describes our design choices and implementation details. Section IV showcases experiments using our modules and presents an analysis of the results to validate their correctness by comparing the observed behavior to analytically-derived expectations. In Section V, we provide instructions to configure these modules in an ns-3 simulation, and finally, we consider future work in Section VI.

II. BACKGROUND

A. Strict Priority Queuing

Strict priority queuing (SPQ) [4] classifies network packets as either priority or regular traffic and ensures that priority traffic will always be served before low priority. Priority packets and regular packets are filtered into separate FIFO queues, the priority queue must be completely empty before the regular queue will be served. The advantage of this method is that high priority packets are guaranteed delivery so long as their inflow does not exceed the transmission rate on the network. The potential disadvantage is a high proportion of priority traffic will cause regular traffic to suffer extreme performance degradation [4]. Figure 1 gives an example of strict priority queuing; packets from flow 2 cannot be sent until the priority queue is completely emptied of packets from flow 1.

B. Weighted Fair Queuing

Weighted fair queuing (WFQ) [5] offers a more balanced approach than SPQ. Instead of giving certain traffic flows complete precedence over others, WFQ divides traffic flows into two or more classes and gives a proportion of the available bandwidth to each class based on the idealized Generalized Processor Sharing (GPS) model [6].

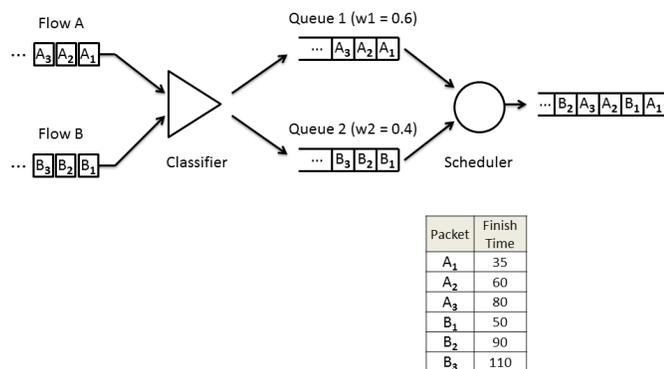


Figure 2. A weighted fair queue

In GPS, each queue i is assigned a class of traffic and weight w_i . At any given time the weights corresponding to nonempty queues w_j are normalized to determine the portion of bandwidth allocated to the queue as shown in (1).

$$w_i^* = \frac{w_i}{\sum w_j} \quad (1)$$

w_i^* is between zero and one and is equal to the share of the total bandwidth allocated to queue i . For any t seconds on a link capable of sending b bits per second, each nonempty queue sends $b * t * w_i^*$ bits.

WFQ approximates GPS by calculating the order in which the last bit of each packet would be sent by a GPS scheduler and dequeues packets in this order [7]. The order of the last bits is determined by calculating the virtual finish time of each packet. WFQ assigns each packet a start time and a finish time, which correspond to the virtual times at which the first and last bits of the packet, respectively, are served in GPS. When the k th packet of flow i , denoted by P_i^k , arrives at the queue, its start time and finish time are determined by (2) and (3).

$$S_i^k = \max(F_i^{k-1}, V(A_i^k)) \quad (2)$$

$$F_i^k = S_i^k + \frac{L_i^k}{w_i} \quad (3)$$

where $F_i^0 = 0$, A_i^k is the actual arrival time of packet P_i^k , L_i^k is the length of P_i^k , and w_i is the weight of flow i . Here $V(t)$ is the virtual time at real time t to denote the current round of services in GPS and is defined in (4).

$$\frac{dV(t)}{dt} = \frac{c}{\sum_{i \in B(t)} w_i} \quad (4)$$

where $V(0) = 0$, c is the link capacity, and $B(t)$ is the set of backlogged connections at time t under the GPS reference system. WFQ then chooses which packet to dequeue based on the minimal virtual finish time. Figure 2 gives an example of weighted fair queuing; packets are sent in the order determined by their virtual finish times.

C. Weighted Round Robin

Weighted round robin (WRR) queuing is a round robin scheduling algorithm that approximates GPS in a less computationally intensive way than WFQ. Every round each nonempty queue transmits an amount of packets proportional to its weight. If all packets are of uniform size, each class of traffic is provided a fraction of bandwidth exactly equal to its assigned weight. In the more general case of IP networks with variable size packets, the weight factors must be normalized using the mean packet size. Normalized weights are then used to determine the number of packets serviced from each queue. If w_i is the assigned weight for a class and L_i is the mean packet size, the normalized weight of each queue is given by (5).

$$w_i^* = \frac{w_i}{L_i} \quad (5)$$

Then the smallest normalized weight, w_{min}^* , is used to calculate the number packets sent from queue i each round as shown in (6) [8].

$$\left\lceil \frac{w_i^*}{w_{min}^*} \right\rceil \quad (6)$$

WRR has a processing complexity of $O(1)$, making it useful for high speed interfaces on a network. The primary limitation of WRR is that it only provides the correct proportion of bandwidth to each service class if all packets are of uniform size or the mean packet size is known in advance, which is very uncommon in IP networks. To ensure that WRR can emulate GPS correctly for variably sized packets, the average packet size of each flow must be known in advance; making it unsuitable for applications where this is hard to predict. More effective scheduling disciplines, such as deficit round robin and weighted fair queuing were introduced to handle the limitations of WRR. Figure 3 gives an example of weighted round robin queuing; because packets sent are rounded up, each round two packets will be sent from flow 1 and one packet from flow 2.

III. RELATED WORK

The predecessor to ns-3, ns-2 [9], had implemented some scheduling algorithms such fair queuing, stochastic fair queuing, smoothed round robin, deficit round robin, priority queuing, and class based queuing as official modules. ns-2 and ns-3 are essentially different and incompatible environments, ns-3 is a new simulator written from scratch and is not an

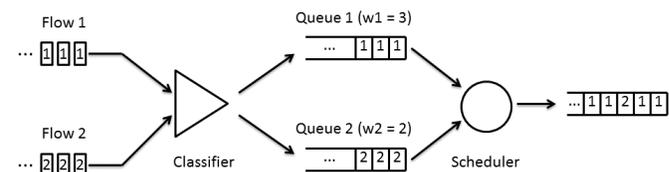


Figure 3. A weighted round robin queue

evolution of `ns-2`. At the time of writing, the latest version, `ns-3.32`, contains no official differentiated service queuing modules. Several modules have been contributed by others, such as a leaky bucket queue implementation [2] and the previously mentioned DiffServ evaluation module [3].

IV. DESIGN AND IMPLEMENTATION

In the DiffServ architecture, there is a distinction between edge nodes, which classifies packets and set the DS fields accordingly, and internal nodes, which queue these packets based on their DS value. In the design framework, we used for all modules, each queue operates independently; we do not utilize the DS field and packets are reclassified at each instance.

WFQ, WRR, and SPQ all inherit from the *Queue* class in `ns-3`. *Queue* provides a layer of abstraction that hides the scheduling algorithm and allows easy utilization of our classes wherever the *Queue* class or any of its inherited classes exist.

The *Queue* API has three main public members related to functionality: *Enqueue()*, *Dequeue()*, and *Peek()*. In the Point To Point module, *PointToPointNetDevice* passes outgoing packets to *Queue::Enqueue()* when it has finished processing them. *PointToPointNetDevice* calls *Queue::Dequeue()* when the outgoing link is free and begins transmitting the returned packet. Our modules were built specifically with Point To Point in mind, but can be included with any `ns-3` module that utilizes *Queue*.

Classes that inherit from *Queue* must implement the abstract methods *DoEnqueue()*, *DoDequeue()*, and *DoPeek()* which are called by the public methods *Enqueue()*, *Dequeue()*, and *Peek()* respectively.

DoEnqueue() takes a packet as an argument, attempts to queue it, and indicates whether the packet was successfully queued or dropped. *DoDequeue()* takes no arguments, attempts to pop the next scheduled packet, and returns the packet if successful or an error otherwise. *DoPeek()* takes no arguments and returns the next scheduled packet without removing it from the queue.

Our classes follow the same functional design pattern: *DoEnqueue()* calls *Classify()* which determines the correct queue based on user provided parameters. *DoDequeue()* and *DoPeek()* both implement the module-specific scheduling algorithm and return the next the scheduled packet.

To handle user provided criteria for classification, we created an input format modeled after Cisco System's IOS configuration commands. For each of our modules, the classifier sorts incoming packets into separate classes based on these criteria: source IP address, source port number, destination IP address, destination port number, and TCP/IP protocol number (TCP or UDP).

The source and destination address criteria could be either a single host or a range of IP addresses. An optional subnet mask can be provided along with the criteria to distinguish the incoming packets from a particular network. We chose to adopt an inverse mask instead of normal mask (0.0.0.255 instead of 255.255.255.0) for consistency with Cisco IOS.

Each set of user-defined match criteria is stored as an Access Control List (ACL). An ACL consists of a set of entries, where each entry is a combination of the mentioned five-tuple values to uniquely identify a group of packets. After ACLs are introduced to the system, each ACL is linked to a CLASS, which matching packets are associated to. Besides an ACL, a CLASS also has attributes such as weight and queue size. Each instance of CLASS must have at least one associated ACL and each ACL can only relate to one class.

Upon arrival of a new packet, our module attempts to classify the packet into an existing CLASS based on ACLs. If a match is found, the packet is placed into the reserved queue for the corresponding CLASS, however if a match is not found, it will be grouped into the predefined default CLASS. Each reserved queue is a first-in first-out queue with a tail drop policy. An example for configuring the ACL input file is included in the usage section.

A. Strict Priority Queuing

1) *Design*: SPQ has two internal queues, Q1 and Q2, Q1 is the priority queue and Q2 is the default queue. A single port or IP address can be set by the user and matching traffic is sorted into the priority queue, all other traffic is sorted into the lower priority default queue.

In some SPQ implementations, outgoing regular priority traffic will be preempted in mid-transmission by the arrival of an incoming high priority packet. We chose to only implement prioritization at the time packets are scheduled.

2) *Implementation*: *DoEnqueue()* calls the function *Classify()* on the input packet to get a class value. *Classify()* checks if the packet matches any of the priority criterion and indicates priority queue if it does or default queue if it does not. The packet is pushed to the tail if there is room in the queue; otherwise, it is dropped.

DoDequeue() attempts to dequeue a packet from the priority queue. If the priority queue is empty then it will attempt to schedule a packet from the regular queue for transmission.

B. Weighted Fair Queuing

1) *Design*: Our class based WFQ assigns each packet a class on its arrival. Each class has a virtual queue with which packets are associated. For the actual packet buffering, they are inserted into a sorted queue based on their finish time values. Class (and queue) weight is represented by a floating point value.

A WFQ's scheduler calculates the time each packet finishes service under GPS and serves packets in order of finish time. To keep track of the progression of GPS, WFQ uses a virtual time measure, $V(t)$, as presented in (4). $V(t)$ is a piecewise linear function whose slope changes based on the set of active queues and their weights under GPS. In other words, its slope changes whenever a queue becomes active or inactive.

Therefore, there are mainly two events that impact $V(t)$: first, a packet arrival that is the time an inactive queue becomes active and second, when a queue finishes service and becomes

inactive. The WFQ scheduler updates virtual time on each packet arrival [7]. Thus, to compute virtual time, it needs to take into account every time a queue became inactive after the last update. However, in a time interval between two consecutive packet arrivals, every time a queue becomes inactive, virtual time progresses faster. This makes it more likely that other queues become inactive too. Therefore, to track current value of virtual time, an iterative approach is needed to find all the inactive queues, declare them as inactive, and update virtual time accordingly [7]. The iterated deletion algorithm [10] shown in Figure 4 was devised for that purpose.

```

while true do
   $F = \text{minimum of } F^\alpha$ 
   $\delta = t - t_{chk}$ 
  if  $F \leq V_{chk} + \delta * \frac{L}{sum}$  then
    declare the queue with  $F^\alpha = F$  inactive
     $t_{chk} = t_{chk} + (F - V_{chk}) * \frac{sum}{L}$ 
     $V_{chk} = F$ 
    update sum
  else
     $V(t) = V_{chk} + \delta * \frac{L}{sum}$ 
     $V_{chk} = V(t)$ 
     $t_{chk} = t$ 
    exit
  end if
end while

```

Figure 4. The iterated deletion algorithm

Here, α is an active queue, F^α is the largest finish time for any packet that has ever been in queue α , sum is the summation of the weights of actives queues at time t , and L is the link capacity.

We maintain two state variables: t_{chk} and $V_{chk} = V(t_{chk})$. Because there are no packet arrivals in $[t_{chk}, t]$, no queue can become active and therefore sum is strictly non-increasing in this period. As a result a lower bound for $V(t)$ can be found as $V_{chk} + (t - t_{chk}) * \frac{L}{sum}$. If there is a F^α less than this amount, the queue α has become inactive some time before t . We find the time this happened, update t_{chk} and V_{chk} accordingly and repeat this computation until no more queues are found inactive at time t .

After the virtual time is updated, the finish time is calculated for the arrived packet and it is inserted into a priority queue sorted by finish time. To calculate the packet's finish time, first its start time under GPS is calculated, which is equal to the greater of current virtual time and largest finish time of a packet in its queue or last served from the queue. Then, this amount is added to the time it takes GPS to finish the service of the packet. This is equal to packet size divided by weight.

2) *Implementation*: Similarly to SPQ's implementation, *DoEnqueue()* calls *Classify()* on input packets to get a class value. *Classify()* returns the class index of the first matching criteria, or the default index if there is no match. This class value maps to one of the virtual internal queues, if the queue is not full the packet is accepted, otherwise it is dropped.

Current virtual time is updated as previously described and if the queue was inactive it is made active. The packet start time is calculated by (2) using updated virtual time and queue's last finish time. Then packet finish time is set by *CalculateFinishTime()*. This method uses (3) to return the virtual finish time. The queue's last finish time is then updated to the computed packet finish time. Finally the packet is inserted into a sorted queue based on the finish numbers. A *priority_queue* from C++ container library was used for that purpose.

DoDequeue() pops the packet at the head of priority queue. This packet has the minimum finish time number.

C. Weighted Round Robin

1) *Design*: WRR has the same number of internal queues, assigned weight representation, and classification logic as WFQ. The weight must be first normalized with respect to packet size. In an environment with variably sized packets, weighted round robin needs to assign a mean packet size s_i for each queue. These parameters are identified by the prior to the simulation in order to correctly normalize the weights. The normalized weight and number of packets sent are calculated by (5) and (6).

2) *Implementation*: Before the start of the simulation, *CalculatePacketsToBeServed()* determines the number of packets sent from each queue using (6). Similar to SPQ and WFQ, *DoEnqueue()* uses *Classify()* to find the class index of the incoming packets and then puts them in the corresponding queue.

DoDequeue() checks an internal counter to track how many packets to send from the queue receiving service. Each time a packet is sent the counter is decremented. If the counter is equal to zero or the queue is empty *DoDequeue()* marks the next queue in the rotation for service and updates the counter to the value previously determined by *CalculatePacketsToBeServed()*.

V. VALIDATION

To validate our WFQ and WRR implementations we ran a series of experiments against each module. For each experiment, we chose a scenario with predictable outcomes for a given set of parameters based on analysis of the scheduling algorithm. Then we ran simulations of the scenario using the module and compared the recorded results with our analytic model.

All our experiments used the six node topology shown in Figure 5. Each sender and receiver pair sends a traffic flow across a single shared link between the two middle boxes. We installed our modules on the outgoing *NetDevice* across the shared link. In order to observe the characteristics of WFQ and WRR, both senders send 1 GB of data in all simulations and the experiments ends after the last packet in either of the flows is received.

The following parameters are constant across our simulations. The senders use the `ns-3` bulk sending application to

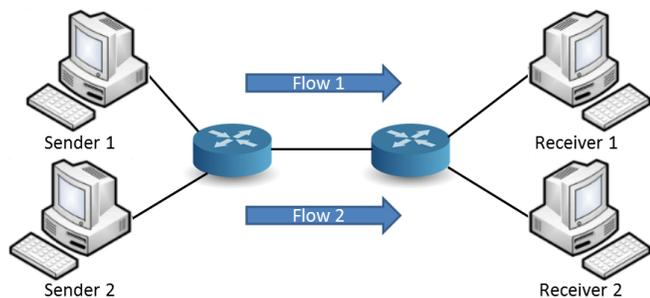


Figure 5. Simulated network used in validation experiments

send 1000 byte packets with no idling, all links have 5ms delay, queue sizes were chosen to be practically unbounded to avoid packet loss.

Because both the weighted fair queuing and weighted round robin are approximations of GPS, and GPS allocates bandwidth based on exact weights, we expect ratio of both traffic flow’s throughput to be close to the ratio of weights.

The two traffic flows are assigned weights w_1 and w_2 where $w_2 = 1 - w_1$ for all simulations. Both traffic flows transmit packets at T Mbps from the senders and the shared link has a throughput capacity of $0.5T$, creating a bottleneck. For each module we ran sets of four simulations where $w_2 = 1, \frac{1}{2}, \frac{1}{7}, \frac{1}{10}$ and T is fixed, we repeated this four times with different data rates $T = 0.5, 1, 10, 50$. In each simulation, the receivers measure the average throughput of both flows, R_1 and R_2 over 1ms intervals and the we record the ratio. All simulations

stopped after the first traffic flow has finished transmitting.

The results in Figures 6, 7, 8, and 9 show the ratio of throughput at the receivers remains close to the ratio of weights. As we increase data rate across the network, the measured ratio converges to the theoretical one.

For each flow in a correctly implemented WFQ system, the number of bytes served should not lag behind an ideal GPS system by more than the maximum packet length. In low data rates such as 0.5Mbps even one packet can make a noticeable difference. For instance, in a GPS system when the ratio of weights is 10, the first flow sends 10 packets and the second flow sends 100 packets over the same time interval. However, in the corresponding WFQ system if the first flow sends 9 packets, then the perceived ratio will be 11.11 instead of 10.

Because WRR is optimal when using uniform packet sizes, a small number of flows, and long connections we observe that WRR performs as well as WFQ in approximating GPS. As expected, we can see in Figures 10, 11, 12, and 13 that the measured ratio of throughput converges to the ratio of weights as data rate increases.

VI. USAGE

CLASSs and their ACLs must be introduced to the simulation before it can begin. Besides working with objects directly in their application, users can provide this information through an XML or text file. In the XML file, using `<class_list>` and `<acl_list>`, a hierarchical structure is specifically designed in which users can enter a list of their CLASSs and ACLs separately. The CLASSs and their corresponding ACLs are

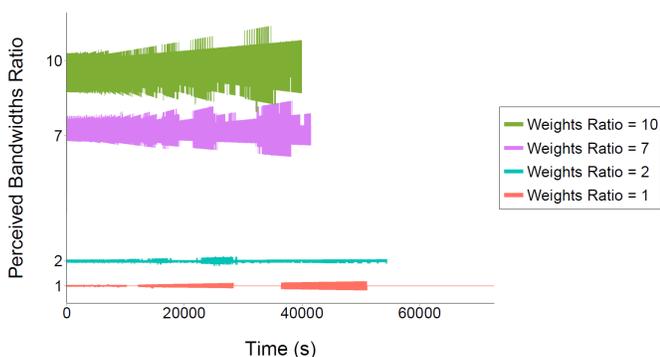


Figure 6. WFQ validation: $T = 0.5$ Mbps

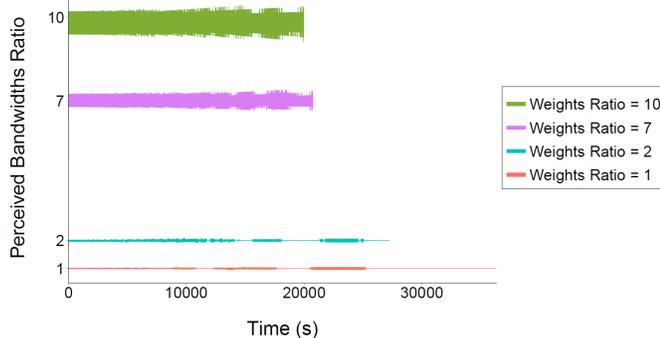


Figure 7. WFQ validation: $T = 1$ Mbps

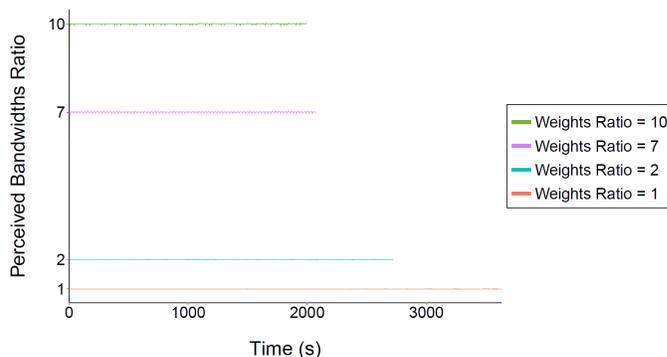


Figure 8. WFQ validation: $T = 10$ Mbps

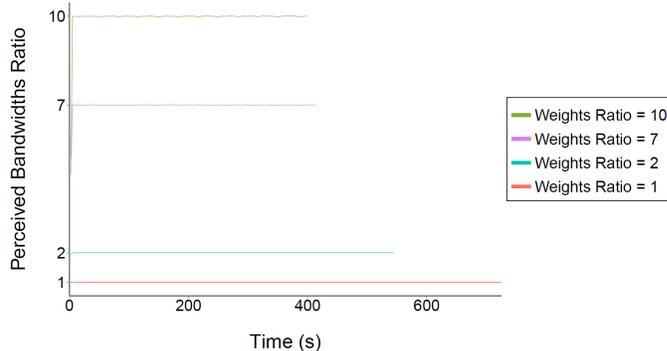


Figure 9. WFQ validation: $T = 50$ Mbps

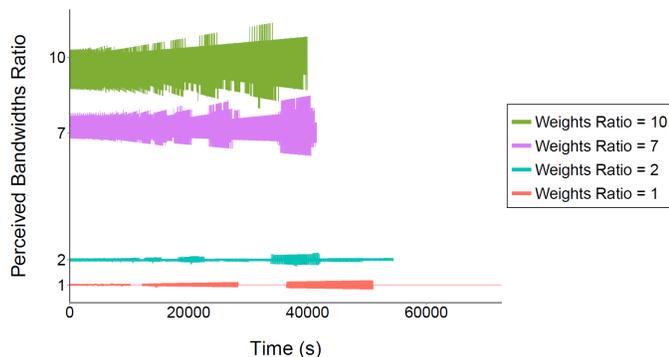


Figure 10. WRR validation: $T = 0.5$ Mbps

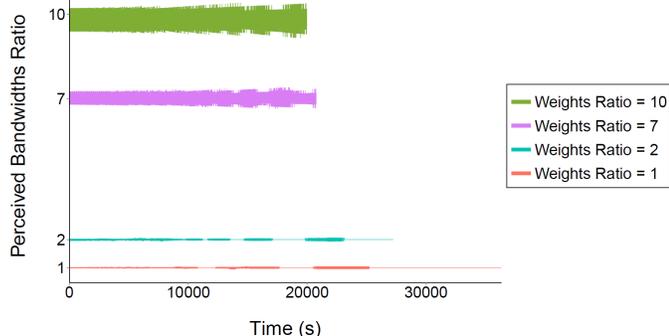


Figure 11. WRR validation: $T = 1$ Mbps

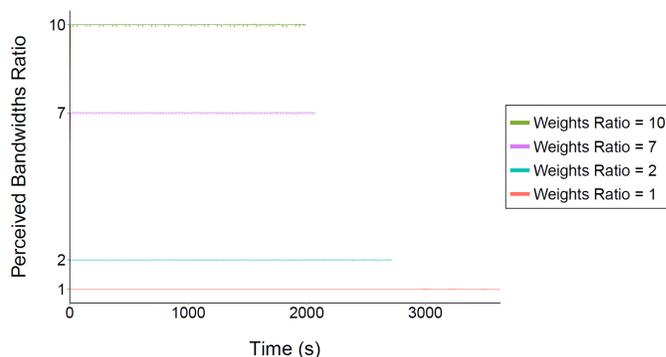


Figure 12. WRR validation: $T = 10$ Mbps

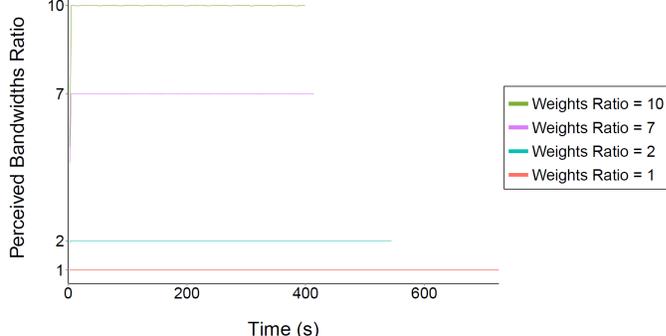


Figure 13. WRR validation: $T = 50$ Mbps

then linked together using `<acl_id>` attribute of the class. We have provided an example similar to what we used in our validation scenario:

Alternatively, users can provide the data through a text file. This file consists of a set of lines where each line is a command designated to introduce an ACL or CLASS to the system. The class and access-list commands are used to define a CLASS and an ACL respectively and the class-map command is used to link the two. These commands are simplified versions of Cisco IOS commands and should be familiar to users who have worked with Cisco products.

VII. CONCLUSION AND FUTURE WORK

In order to add new functionality to `ns-3`, we have designed and implemented modules for strict priority queuing, weighted fair queuing, and weighted round robin. We have described how these modules correctly implement their respective algorithms within the `ns-3` framework and left the reader with means to utilize them for further experimentation. The ease of configuration and use of our modules should make them attractive tools for further research and we look forward to seeing how others take advantage of our work.

There is a large amount of overlapping functionality between the three queues, particularly WFQ and WRR. All three modules perform basic classification and scheduling at the same points and some of this functionality could be combined into a shared base class for different types differentiated service queues. A stateful classifier and scheduler could be

```
<acl_list><acl id=aclFirstClass><entry>
  <source_address >10.1.1.0</source_address>
  <source_address_mask >0.0.0.255</source_address_mask>
  <source_port_number >23</source_port_number>
  <destination_address >172.16.1.0</destination_address>
  <destination_address_mask >0.0.0.255</destination_address_mask>
  <destination_port_number >23</destination_port_number>
  <protocol>TCP</protocol>
</entry></acl></acl_list>
```

```
<<class_list>
  <class id="class1" acl_id="aclFirstClass">
    <queue_size >256</queue_size>
    <weight >0.875</weight>
  </class>
</class_list>
```

```
access-list access-list-id [protocol] [source_address] [source_address_mask]
 [operator [source_port]] [destination_address] [destination_address_mask]
 [operator [destination_port]]
access-list aclFirstClass TCP 10.1.1.0 0.0.0.255 eq 23 172.16.1.0 0.0.0.255 eq 23
```

```
class [class_id] bandwidth percent [weight] queue-limit [queue_size]
class class1 bandwidth percent 0.875 queue-limit 256
```

```
class-map [class_id] match access-group [acl_id]
class-map class1 match access-group aclFirstClass
```

Figure 14. `class_list`, `acl_list`, `access-list`, `class`, and `class-map`

implemented in a child class or associated with the base class as part of a framework for creating these queues.

REFERENCES

- [1] "The ns-3 Network Simulator," Project Homepage. [Online]. Available: <http://www.nsnam.org> [Retrieved: September, 2015]
- [2] P. Baltzis, C. Bouras, K. Stamos, and G. Zaoudis, "Implementation of a leaky bucket module for simulations in ns-3." tech. rep., Workshop on

- ICT - Contemporary Communication and Information Technology, Split - Dubrovnik, 2011.
- [3] S. Ramroop, "Performance evaluation of diffserv networks using the ns-3 simulator," tech. rep., University of the West Indies Department of Electrical and Computer Engineering, 2011.
 - [4] Y. Qian, Z. Lu, and Q. Dou, "Qos scheduling for nocs: Strict priority queueing versus weighted round robin," tech. rep., 28th International Conference on Computer Design, 2010.
 - [5] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single node case," *IEEE/ACM Transactions on Networking*, vol. 1, no. 3, 1993, pp. 344-357.
 - [6] A. Demers, S. Keshav, and S. Shenker, "Analysis and simulation of a fair queueing algorithm," *ACM SIGCOMM*, vol. 19, no. 4, 1989, pp. 3-14.
 - [7] S. Keshav, *An Engineering Approach to Computer Networking*. Addison Wesley, 1998.
 - [8] M. Katevenis, S. Sidiropoulos, and C. Courcoubetis, "Weighted round-robin cell multiplexing in a general-purpose atm switch chip," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 8, 1991.
 - [9] "The ns-2 Network Simulator," Project Homepage. [Online]. Available: <http://www.isi.edu/nsnam/ns/> [Retrieved: September, 2015]
 - [10] S. Keshav, "On the efficient implementation of fair queueing," *Journal of Internetworking: Research and Experience*, vol. 2, no. 3, 1991.

Simulation of the Influence of Increased Speed Limit and Densification of Stations on Travel Time Based on the Example of the First Metro Line in Warsaw

Marek Bauer

Department of Transportation Systems
Cracow University of Technology (PK)
Cracow, Poland
e-mail: mbauer@pk.edu.pl

Abstract—The author’s original model for travel time of metro trains is presented in this paper. The presented model is assumed to be an instrument for estimation of reliable travel times of metro trains in macro simulation models of whole transport networks. These kinds of models are used to evaluate the effects of the possible implementation of major transport investments in cities. The quality of the network model can significantly affect the results of analyses, and even the selection of optimal solutions. Travel time is the basis for every transport network model, not only for the current state, but also for the future. This is particularly important in the case of planned sections, where one cannot use measurement results. Partial models of metro average running times and random components of running time, depending on the length of sections, were created on the basis of the results of measurements carried out for the first metro line in Warsaw. Also, the dwell time at stations is included into the model based on the results of measurements made for three separate station types, diversified on the basis of average stopping times and location of the station in relation to the inner city centre. The paper includes examples of practical model application: evaluation of the influence of increasing the allowable speed, and densification of the stations on travel time and also the impact of the accuracy of the travel time model on simulation results of passenger flows.

Keywords—metro; traffic simulation; travel time.

I. INTRODUCTION

The metro is commonly perceived as one of the best urban transport systems, as it enables fast transportation of passengers. Its utility increases appropriately to the distance of a trip. This is the effect of consequent separation of the entire track, and long distances between stations, which allows high speed train, up to 60 [km/h] or more. This results in high transportation speeds (compared and summarised by Feng at al. [6]), of around 35-40 [km/h]. The metro is also associated with a large transport capacity [13] [17], as well as high operational reliability [3]. However, high speed does not go hand in hand with accessibility. Long distances between stations lead to long walking distances to find a station. These issues have been discussed by García-Palomares at al. [7]. For this reason, a metro does not perform well in high density areas. Significant investment and operating costs are another downside of the metro.

The aforementioned pros and cons create the necessity to perform a thorough analysis of costs and benefits before the potential introduction or expansion of a metro system. Discussions as to, whether it is legitimate to build (or extend) a metro line are usually a consequence of the insufficient utility of existing systems of urban public transport. Very often, this is a common effect of a lack of determination and a consequence of giving priority to over-ground means of public transport. Arguments used in these discussions are very different, but always one of them is high travel speed of metro trains.

Discrepancies between expected and possible travel times, appear as early as during the planning stage. Meanwhile, travel time has a crucial influence on modal split, especially considering rail and tramway systems that are able to transport the greatest passenger volumes. Even slight differences in travel time may have a considerable influence on increased or decreased passenger volumes, not only on the metro line itself, but also on other lines of public transport. These differences also result in changes in car volumes.

Therefore, travel time is an obligatory element of a four-stages procedure (described, among others, by Hensher [9]), commonly used in the planning of transportation systems. Reliable estimation of transportation speed for individual variants of metro line routes is obligatory for a determination of potential effects of differentiated variants of transport investments. Only such an approach allows a reliable determination of outcomes of serious transport investments.

To provide a fully reliable basis for the potential construction of a metro line, it is necessary to describe processes of train movement in greater detail. It is thus essential to possess precise knowledge of both current and achievable realistic running times and times spent by trains at the stations.

This paper presents a general travel time model for a metro line that may be used for planning such investments in various cities. The main purpose of the analysis was to recognise the mechanism of changes in travel time of metro trains and to build a model that would serve to estimate the length and variability of travel time. Section 2 presents a general model of travel time for a metro line, whilst in Section 3 there is a description of measurement’ results from the first metro line in Warsaw. Sections 4 and 5 explain the results of estimations of running time and dwell time at the

stations. In Section 7, examples of possible simulation results are shown. Finally, Section 8 presents the conclusion of the paper.

II. GENERAL MODEL OF TRAVEL TIME FOR A METRO LINE

A. Structure of the model of travel time for a metro line

The graph theory [8] allows the assumption that consecutive stations are vertices of a directed graph and the sections between the stations are directed edges. From the point of view of a general structure, the model of travel time for a metro line will not differ from travel time models for other means of transport, e.g., trams or buses, described for instance by Vuchic [12] and Bauer [2].

However, there are certain differences at the stage of modelling events specific to a metro line. In models of bus or tram lines, the registered events are: arrivals and departures at stops, moments of starting and ending the process of passenger alighting and boarding, as well as the moments of standstill at any given point within the network, such as crossroads or pedestrian crossings. In a model of a metro line, the number of occurrences may be significantly limited [4]. The fully separated route of a metro line and the full (or at least very advanced) automation of train rides decrease the need to consider any events between stations. Of course, such events may happen (mostly for security and safety reasons), but they are relatively rare. In turn, automated processes of simultaneous opening or closing of train doors question the division of the dwell time into separated processes (alighting and boarding, time of waiting for possibility to depart). As a result, only two types of events on the line have been accounted for in the presented model:

- the moment when a train is brought to a halt upon arrival in a station (defined as the moment of starting of opening the door);
- the moment when the train leaves the station (the same as moment of starting of closing the door).

Thus, it may be acknowledged that the structure of the travel time model for a metro line will comprise a group of modules: section between stations - station at the end of section. Only two processes of movement will be considered:

- running time – counted from the moment of departure from a station until arrival at the next station;
- dwell time – time spent at a station, counted from the moment of arrival to the moment of departure.

The analysis omitted time spent at the first and last station, as these do not have any direct influence on train travel time and the duration of passengers' trips.

B. Mathematic structure of the model of travel time for a metro line

The mathematical structure of the proposed model strictly corresponds to its approved general structure. In general, travel time for trains within a metro line is the sum of travel times for sections between stations and the time spent at the stations [1]:

$$T = \sum_{m=1}^p t_{r,m} + \sum_{m=1}^{p-1} t_{s,m} \quad (1)$$

where:

T – travel time for the whole metro line (one direction) [s];
 $t_{r,m}$ – running time for an individual section between stations (v_{m-1}, v_m) [s];
 $t_{s,m}$ – dwell time - time spent during a halt in a station m [s].

According to the deterministic approach currently used to solve most practical problems, average travel time for a line may be presented as the sum of average values of travel time and dwell time:

$$\bar{T} = \sum_{m=1}^p \bar{t}_{r,m} + \sum_{m=1}^{p-1} \bar{t}_{s,m} \quad (2)$$

However, in more detailed analyses it is desirable not only to decrease average travel time, but also to decrease dispersion of travel times for a line. This justifies the consideration of additional random components of running time and dwell time. It is thus reasonable to expand the formula (1) by random components, as presented:

$$\tilde{T} = \sum_{m=1}^p (\bar{t}_{r,m} + \hat{t}_{r,m}) + \sum_{m=1}^{p-1} (\bar{t}_{s,m} + \hat{t}_{s,m}) \quad (3)$$

The ability to estimate travel time dispersion increases the quality of the model and spreads the scope of its uses such a model can also be used in dynamic modelling. Taking well-adjusted random components of the model, one can calculate the influence of travel time variability on passenger volumes.

III. FIRST LINE OF THE WARSAW METRO

A. Description of the metro line

Estimation of travel times for sections between stations and dwell time at the stations were based on results of automatic traffic measurements on the first metro line in Warsaw. The line is 21.648 [km] long (one direction) with 21 train stations. Distances between stations [18] vary from 0.577 to 1.534 [km], with an average distance of 1.082 [km]. The line constitutes a vital element of the track-based urban transportation system of Warsaw. For this reason, it has been decided to build a second metro line in the East – West direction [16].

According to [15], an estimated 147.2 million passengers used the Warsaw metro in 2013. Metro trains covered a total of 23.8 thousand [train-km] during this period.

B. Transportation speed

The calculations used measurement results made available by the Public Transport Authority for Warsaw. The results are from two average working days (19.11.2013 and 16.01.2014, when no serious traffic disturbances were observed). The analysis encompassed 581 train journeys (in

the direction “Kabaty” – “Młociny”) and 591 train journeys (in the direction: “Młociny” – “Kabaty”) that occurred between 4 AM and 12 PM. Even though the results used concerned 2 days only, they have been acknowledged as being representative for an average working day on the Warsaw metro. Transportation speeds on the first metro line are considerably high, with a daily average of 35.9 [km/h] (direction: “Kabaty” – “Młociny”) and 35.4 [km/h] (direction: “Młociny” – “Kabaty”). This does not mean that the transportation speeds are fixed. On the contrary, differences between speeds in the following hours of the analysis may be significant, as illustrated in Figure 1.

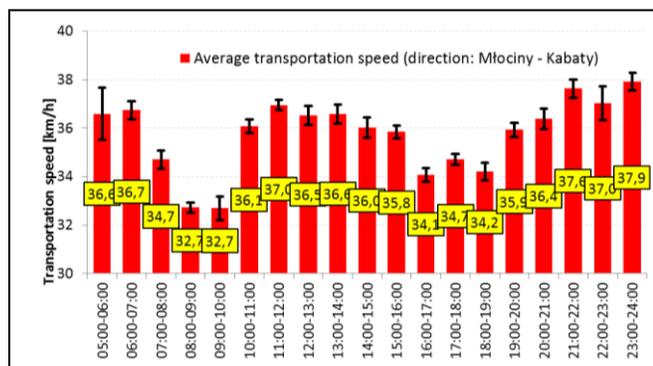


Figure 1. Variability of transportation speeds on the first metro line (direction: Młociny - Kabaty) within the hours of the line operation.

This figure shows the scale of standard errors connected with estimation of average transport speeds. In travel time modelling, it is vital to reproduce traffic conditions within a given time of day. Dispersion of transportation speeds occurs even on metro lines. Thus, it has been suggested to divide a working day into six time periods, differing mostly in term of current train frequency in the first metro line in Warsaw:

- **early morning** – train departures within the hours: 04:00 AM - 5:59 AM, estimated scheduled interval on the line: 4-7 [min];
- **morning** – departures: 6:00 AM - 8:59 AM, interval: 3 [min];
- **late morning and early afternoon** – departures: 9:00 AM - 02:59 AM, interval: 4 [min];
- **afternoon** – departures: 03:00 PM - 06:59 PM, interval: 3 [min];
- **early evening** – departures: 07:00 PM - 08:59 PM, interval: 4-7 [min];
- **evening** – departures: 09:00 PM - 11:59 PM, interval: 9-10 [min].

Usually, models of transport networks are built for the morning or afternoon peak hours, when most journeys happen. On the first metro line, average travel time during the afternoon peak is 37.5 [min] (difference between average travel times in both directions is only 3 [s]). This means that during this period, transportation speed equals 34.7 [km/h].

Models for other time periods can also be created. It is most important to adjust the model of travel time on the metro line to the models of travel time of other means of

transport, so as to ensure realistic effects in terms of passenger flow between the various means of transport. That is why the aforementioned division of the working day into time periods has been proposed. Relatively speaking, the highest values of average transportation speed were reached within the hours between peaks, in early evening and evening hours - above 36 [km/h]. Lowest average transportation speeds were reached during single hours of the morning peak, between 08:00 AM and 09:00 AM and from 09:00 AM to 10:00 AM (direction: “Młociny” – “Kabaty”) and the afternoon peak, from 05:00 PM to 06:00 PM (second direction). Within the aforementioned hours, average transport speeds fluctuated around 33 [km/h].

Variability of transportation speeds within a single hour was highest in the morning and evening periods. This may be caused by the lower number of train courses and greater size diversification of passenger flow. The results confirm the fact that, despite the total separation of the metro system from other means of transport within the traffic system, variability in transportation speeds occurs and it is worthwhile considering the reasons for this.

C. Travel time structure of the first metro line

The search for the reasons for dispersed travel times of metro trains began with a division into running times between stations and dwell times. This allowed the determination of which element of the metro line was more susceptible to disturbances. The ratio of the total sum of running times to whole travel time is 67.5 [%] (direction: “Kabaty” – “Młociny”) and 68.3 [%] (“Młociny” – “Kabaty”). This means that an average of 1/3 total travel time for the entire line is spent at stations. This ratio changes from 67.5 [%] in the afternoon (direction: “Kabaty – Młociny”) up to as much as 72.8 [%] in the evening (“Młociny – Kabaty”).

D. Comparison of changes in time of ride between stations and time spent during halts in the stations

The longest travel times for the metro line have been observed in the morning and afternoon periods. In general, slightly greater differences have been observed in the direction “Młociny” – “Kabaty”, as average travel time in the early evening is 3.0 [min] shorter than the corresponding time in the morning (Figure 2).

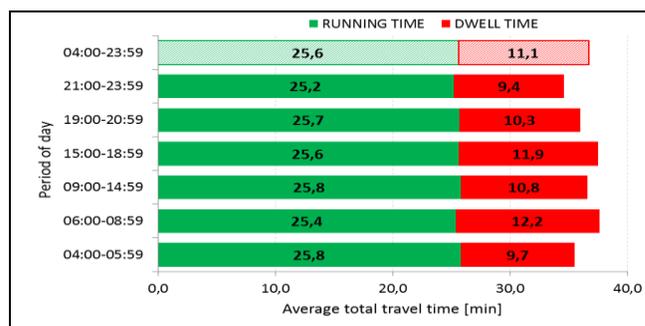


Figure 2. Variability of total travel time divided into running time between stations and dwell time (direction: “Młociny” – “Kabaty”).

In the other direction, the difference is 2.7 [min] – between the early evening period, when travel time is shortest, and the afternoon period of the day. It may also be observed, that differences in running times are significantly smaller than differences between dwell times. This fact will be taken into account during model component estimation.

IV. ESTIMATION OF TRAVEL TIME FOR A SECTION BETWEEN STATIONS

A. Average running time

Running time between stations - in the case of a metro system - is affected by considerably fewer factors than in the case of over-ground systems of urban public transport. Stop-to-stop sections on the metro line are fully separated, the most crucial factor affecting the travel time is section length. Far less important are geometrical factors such as horizontal curves or longitudinal grades. Possible travel time disturbances usually result from the necessity to counteract atypical situations. For example, longer dwell time of preceding train leads to slowing down or even stopping the upcoming trains between the stations. Differences in running time can also be caused by dispatching tactics, especially in the case of a high frequency of trains. These influences were described by Yang et al. [14] and also by O. Yalçınkaya and G.M. Bayhan [13]. Important is also the condition of a track system and the kind of power system.

First, the estimation of the average running time for a section between stations was carried out. The aim was to check how the length of a section influences the running time and, if a satisfactory model was not obtained, other factors were taken into consideration. The metro has an automatic system control, so a physical model, based on calculations of acceleration and deceleration times and running time with constant speed, can be used:

$$t_r = \frac{L}{v_{max}} + \frac{v_{max}}{2} \left(\frac{1}{a} + \frac{1}{b} \right) \quad (4)$$

where:

t_r – average travel time for a section between stations [s];

L – length of a section between stations [km];

v_{max} – maximum speed on the section [m/s];

a – average acceleration rate [m/s²];

b – average deceleration rate [m/s²].

After preliminary application of the model, it transpired that the differences between the measurement results and the calculated results were relatively large - a total of nearly one minute in one direction. Therefore, a statistical, regression model was sought, which would give a better fit to the measurement results and, additionally, be more versatile and able to take into account random influences. For this reason, the linear and non-linear regression methods were used.

In the case of four out of the six periods of a working day, the best adjustment of the model of average running time to the results of measurements carried out for the first

metro line in Warsaw was obtained through the use of the double-squared model:

$$\bar{t}_r = \sqrt{(\beta_0 + \beta_L \cdot L^2)} \quad (5)$$

where:

t_r – average travel time for a section between stations [s];

L – length of a section between stations [km];

β_0, β_L – simple regression parameters [-].

In the two remaining cases, other models offered a slightly better fit (1-2 [%] higher values of determination coefficients). Eventually, it was decided to use the double-squared model for all the periods of a day. Figure 3 shows the effects of fitting the model to the measurement results.

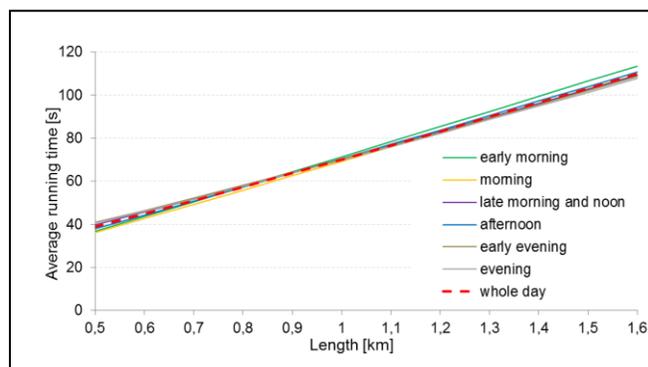


Figure 3. Models of average running time dependent on section length, for acknowledged periods of a working day.

Comparing models for particular periods of a working day, it may be noticed that there are only slight differences in the obtained average travel times. For example, for a section of 0.7 [km] the maximum difference equals just 3 [s] – running time in the morning equals 49 [s], while around noon, in the afternoon and in the early evening it amounts to 52 [s]. For a section of 1.5 [km], average running times differ by 6 [s] – from 101 [s] in the evening to 107 [s], early in the morning. Such a small variability of average running times encouraged the search for a single universal formula for the whole working day. This has the following form:

$$\bar{t}_r = \sqrt{(363 + 4564 \cdot L^2)} \quad (6)$$

This dependence is characterised by a high value for the determination coefficient, amounting to 98 [%]. This means that the length of a section between stations can be considered as the only crucial factor influencing the average travel time on this section, regardless of the period of the working day. It also means that the frequency of train runs even as high as 20 trains per hour does not influence the average travel time on the section.

This model of average running time can be easily used during the construction or actualisation of the Warsaw simulation transportation model. With a couple of additional

assumptions, it can also be useful in the transportation models of other Polish cities, where maximum running speed of trains will be 60 [km/h].

Conversion factors that may be applicable in the case of modelling higher maximum running speeds were also calculated. It was established that only the section length on the metro line has a crucial meaning for running speed. In this calculation (because of a lack of data), physical dependences were used. Maximum running speeds of 70, 80 and 90 [km/h] were taken into consideration. In all three cases, one can use the regression formula:

$$k = \sqrt{\eta_0 + \frac{\eta_1}{L}} \tag{7}$$

where:

k – conversion factor [-];

L – length of a section between stations [km];

γ_0, γ_1 – regression parameters [-].

Values of regression parameters for equation (7) are shown in table I.

TABLE I. REGRESSION PARAMETERS FOR RECALCULATION OF RUNNING TIMES

Regression parameters [-] for different increase of speed					
60 -> 70 [km/h]		60 -> 80 [km/h]		60 -> 90 [km/h]	
γ_0	γ_1	γ_0	γ_1	γ_0	γ_1
0.756	79.9	0.593	143.4	0.477	197.9

B. Random component of running time

In the case of micro-simulation models, there is a possibility of considering the randomness of travel time. Then, it becomes necessary to generate random travel time coefficients for particular sections. These can be simulated by a random number generator from normal distribution with mean zero and standard deviation specified on the basis of measurement results (in [s]):

$$\hat{t}_r = N[0; \sigma_r] \tag{8}$$

Values for standard deviations of the running time can be determined as a function of a section's length. The best fit of the model to the measurement results is as follows:

$$\sigma_r = e^{(\gamma_0 + \gamma_1 \cdot L^2)} \tag{9}$$

where:

σ_r - standard deviation of running time for a section between stations [s];

L – length of a section between stations [km];

γ_0, γ_1 – regression parameters [-].

Initially, partial models were constructed for all of the six periods of a working day (Figure 4).

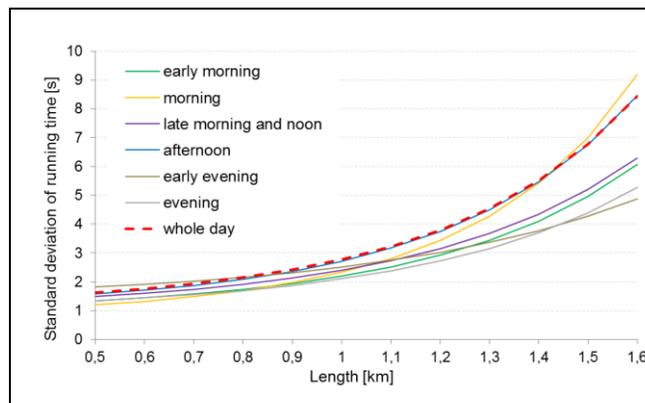


Figure 4. Models of standard deviation of running time dependent on section length, for acknowledged periods of a working day.

However, standard deviations of running time, obtained thanks to the models, differ very little from one another. For sections of lengths up to 1.2 [km], the difference amounts to a maximum of 1 [s], while in the case of a section of 1.6 [km] – only 4 [s]. This is why only one formula was chosen, and this describes the relation between the standard deviation of a running time and the section's length. The formula is representative for a whole working day:

$$\sigma_r = e^{(0,310 + 0,712L^2)} \tag{10}$$

The model above is not characterised by a high value for the determination coefficient, as it is in the case of the model for the average value – R^2 amounts to only 48 [%]. However, taking into consideration the fact that in the case of the first Warsaw metro line the standard deviations are an order of magnitude smaller than average values – the model can be considered satisfactory, thus it can be recommended to be used in micro-simulation analyses.

V. ESTIMATION OF DWELL TIME OF A METRO TRAIN

A. Average dwell time

The dwell time has a more individualised character. It depends mainly on the size of flow of alighting and boarding passengers, and on the number of passengers already travelling on a train arriving at a station. In the case of new investments, the flows are determined at the stage of network planning. It is necessary to determine preliminary dwell times, which may be later verified by an iterative procedure. Since at the stage of carrying out the analyses, reliable sizes of passenger flows for the first line of Warsaw metro were not available, it was decided to differentiate the stations on the basis of the average dwell time. Interestingly, the dependence between the length of dwell time and the location of the station relative to the inner centre of the city was observed. On this basis, 3 types of stations were defined:

- Type A – stations located in the inner city centre (in the case of the first line of the Warsaw metro – four

stations in both directions: “Ratusz Arsenal”, “Świętokrzyska”, “Centrum” and “Politechnika”);

- Type B – stations located within the urban area (“Plac Wilsona”, “Dworzec Gdański”, “Pole Mokotowskie”, “Raclawicka”, “Wierzbno” and Wilanowska);
- Type C – stations located in the remaining areas of the city (remaining stations on the first metro line).

Table II shows the breakdown of average values of dwell time at station types A, B and C.

TABLE II. AVERAGE VALUES OF DWELL TIME DEPENDENT FROM THE LOCATION OF THE STATION RELATIVE TO THE INNER CENTRE

Type of station	Average dwell time [s]						
	Early morning	Morning	Late morning and noon	After-noon	Early evening	Evening	Whole day
A	36	47	39	50	38	34	43
B	31	38	34	39	33	31	36
C	28	32	31	32	30	29	31
All stops	31	37	34	38	33	30	35

It can be seen that the values of average dwell time vary significantly during the day. This variability should be taken into account in macro-simulation models performed for peak periods. In a more specific approach, average dwell time can also be estimated more precisely as a function of numbers of alighting and boarding passengers – if this kind of data is available.

B. Random component of dwell time

Dwell time generally has a Normal distribution, which is confirmed by the Chi-square statistical test. Therefore, the random components of dwell times (in seconds) can be determined from the Normal distribution of zero mean value:

$$\hat{t}_s = N[0; \sigma_s] \quad (11)$$

Standard deviations were calculated for three specified types of stations: A, B and C (Table III).

TABLE III. DWELL TIME STANDARD DEVIATIONS DEPENDENT FROM THE LOCATION OF THE STATION RELATIVE TO THE INNER CENTRE

Type of station	Standard deviation of dwell time [s]						
	Early morning	Morning	Late morning and noon	After-noon	Early evening	Evening	Whole day
A	4	12	6	11	6	4	7
B	4	5	3	6	4	3	4
C	1	2	2	3	3	2	2
All stops	4	8	4	9	5	3	6

A much greater diversity of standard deviations for dwell time can be seen than in the case of running time. This is due to the fact that the number of alighting and boarding passengers is very random.

VI. EXAMPLES OF THE MODEL IMPLEMENTATION

A. Considered scenarios

Implementation of the metro line travel time model has been shown on the basis of the analysis of possibility of densification of Warsaw’s first metro line stations and led to the announcement of raising the maximum speed to 80 [km/h]. The first one is associated with a lengthening of running time, the second with its shortening. The following scenarios are considered:

- **Scenario S0:** existing state of first metro line;
- **Scenario S1:** two additional metro stations (both directions): station “Plac Konstytucji” (Type A, between existing stations: “Politechnika” and “Centrum”) and station “Muranów” (Type B, between stations: “Ratusz Arsenal” and “Dworzec Gdański”) and changing the type of the station “Ratusz Arsenal” from A to B;
- **Scenario S2:** increasing the maximum speed of trains from 60 to 80 [km/h];
- **Scenario S3:** two additional metro stations: “Plac Konstytucji” (Type A) and “Muranów” (Type B), changing the type of the station “Ratusz Arsenal” from A to B, and additionally increasing the maximum speed of trains from 60 to 80 [km/h].

In this calculation, only average metro trains travel times in afternoon period were calculated. Additionally, it was stated that passengers streams were constant, and dwell times would be the same. An extended analysis should take into account the influence of growth of passenger streams because of station densification in the city centre and growth of running speeds on the sections.

B. Comparison of travel times

The model was positively verified. In both directions, there are only 12 [s] differences between total travel time from the model and from the measurements. This means that the model can be used to calculate the effect of increasing the running speed and implementation of new stations.

The final results of the calculations of travel times are presented in Table IV. Introduction of two additional stations will result in lengthening of the average travel time by 1.4 [min], which will lead to a decrease in the average transportation speed from 34.9 to 33.5 [km/h]. However, increasing the running speed from 60 to 80 [km/h] can compensate for this inconvenience. After scenario S3 implementation, total travel time decreases to 35.2 [min] in comparison to the current state, which is a very satisfying value. More precise effects can be specified thanks to the iteration procedure (macro-simulation analysis), where the above values should be used as initial values. Only a multi-criteria analysis will allow full assessment of the options, e.g., via the method described by Nosal and Solecka [10].

TABLE IV. COMPARISON OF TRAVEL TIMES

Parameter of the metro line	Scenario			
	S0	S1	S2	S3
Total running time [min] (all sections)	25.23	25.41	21.49	21.84
Total dwell time [min] (all stations)	12.03	13.33	12.03	13.33
Total travel time [min]	37.26	38.74	33.52	35.17
Share of total running time in total travel time [-]	0.677	0.656	0.641	0.621
Transportation speed [km/h]	34.9	33.5	38.7	36.9
Change of transportation speed, in relation to S0 [%]	*	4.0	-10.9	-5.7

C. The effects of travel time model in the simulation of the entire transport network of Warsaw

Comparisons were made for passenger flows for the scenario S3 using the simulation model of the Warsaw transport network. To estimate the travel time of metro trains, a common approach of constant running speed was used, together with a proposed approach: a model of travel time, which takes into account the variability of travel time. The calculations were performed for the years 2016 and 2036. The results are shown in Table V.

TABLE V. COMPARISON OF PASSENGER STREAMS – S3

Parameter of the metro line	Constant running speeds		Speeds from the presented model	
	2016	2036	2016	2036
Number of metro passengers	53 877	61 745	54 119	62 189
Number of all public transport passengers [persons/h]	321 952	406 817	322 312	407 287
Total number of passengers (public and private transport) [persons/h]	418 610	542 083	418 610	542 083

A more detailed approach resulted in a more accurate estimate of the number of users of the entire system of public transport. As a result, the process of calibration of the Warsaw simulation model was much easier.

VII. CONCLUSIONS

Even a highly automated urban transport system such as a metro is characterised by a certain variability of travel time. What can be observed is a decrease in transportation speeds during peak hours and speed increase in the evenings. Average running times and random components of running times on the sections between stations can be very precisely estimated on the basis of their length, regardless of the period of the day. The frequency runs up to as high as 20 trains per hour does not significantly influence the average running time. However, stations stops definitely have a greater influence on the variability of travel time. Average values for dwell times and their random components were determined for 3 basic station types, dependent on the location of the station relative to the inner centre.

The presented model for the travel time of the metro line, can be used as an input to the macro-simulation models of cities with metro systems. The results of parameter estimation of the model are of local character and depict current and future operational conditions of the first metro line in Warsaw. However, they can be adjusted to local conditions in other cities.

Further work on the halt times will aim at including the size of passenger flows.

REFERENCES

- [1] M. Bauer, "Underground travel time estimation for macro simulation models, on the example of first line in Warsaw," *Budownictwo i Architektura*, 13(4), 2014, pp. 67-74.
- [2] M. Bauer, "Simulation Model of a Bus Line in Changing Traffic Conditions," *The Fifth International Conference on Advances in System Simulation (SIMUL 2013) IARIA*, Nov. 2013, pp. 29-35, ISBN: 978-1-61208-308-7
- [3] L. Castaneda and B. Żótkowski, "Multidimensional system for evaluating the safety and comfort travelling on railway vehicles," *Diagnostyka*, Vol. 2(46), 2008, pp. 45-50.
- [4] S. Derrible and Ch. Kennedy, "Network Analysis of World Subway Systems Using Updated Graph Theory," *Transportation Research Record*, Vol. 3, 2009, pp. 17-25.
- [5] B. S. Everitt and A. Skrondal, "The Cambridge dictionary of statistics," Cambridge, 2002.
- [6] X. Feng, B. Mao, X. Feng, and J. Feng, "Study on the maximum operation speeds of metro trains for energy saving as well as transport efficiency improvement," *Energy*, *The International Journal*, Vol. 36, Issue 11, 2011, pp. 6577-6582.
- [7] J. C. García-Palomares, J. Gutiérrez, and O. D. Cardozo, "Walking accessibility to public transport: an analysis based on microdata and GIS," *Environment and Planning B: Planning and Design*, 40 (6), 2013, pp. 1087-1102.
- [8] M. C. Golumbic, "Algorithmic graph theory and perfect graphs," North Holland, 2004, eBook ISBN: 9780080526966
- [9] D. Hensher and K. J. Button, "Handbook of Transport," Pergamon Press, 2005.
- [10] K. Nosal and K. Solecka, "Application of AHP Method for Multi-criteria Evaluation of Variants of the Integration of Urban Public Transport," *Transportation Research Procedia*, Vol. 3, 2014, pp. 269-278, doi:10.1016/j.trpro.2014.10.006
- [11] Transport for London: Rolling stock data sheet, London 2007.
- [12] V. R. Vuchic, "Urban Transit Systems and Technology," John Wiley & Sons, 2007.
- [13] O. Yalçınkaya and G. M. Bayhan, "Modelling and optimization of average travel time for a metro line by simulation and response surface methodology," *European Journal of Operational Research*, Vol. 196, Issue 1, 2009, pp. 225-233.
- [14] X. Yang, X. Li, Z. Gao, and H. Wang, "A Cooperative Scheduling Model for Timetable Optimization in Subway Systems," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 14, 2013, pp. 438-447.
- [15] <http://www.metro.waw.pl/pliki/PDFy/raport%20roczny%20013%20mw.pdf> [10, 2015]
- [16] <http://www.metro.waw.pl/ii-linia-metra-109.html> [10, 2015]
- [17] <http://www.siemens.com/press/innotrans2012> [9, 2015]
- [18] <http://www.ztm.waw.pl/mapy.php?i=14&c=117&l=1> [10, 2015]

Upgrade of the MovSim for Easy Traffic Network Modification

Andraž Omahen*, Teo Kukuljan†, Vida Vukašinović‡, Peter Korošec‡ and Gregor Papa‡

*Faculty of Computer and Information Science, Ljubljana, Slovenia

Email: andraz.omahen@gmail.com

†Faculty of Mathematics and Physics, Ljubljana, Slovenia

Email: teo.kukuljan@student.fmf.uni-lj.si

‡Computer Systems department, Jožef Stefan Institute, Ljubljana, Slovenia

Email: {vida.vukasinovic, peter.korosec, gregor.papa}@ijs.si

Abstract—Microscopic traffic models describe the individual elements of transportation system, such as the motion of individual vehicle. In this paper, MovSim, an open-source tool for model investigation and simulation, is used and upgraded as a comprehensive tool for building microscopic traffic models. The upgraded simulator allows the construction of complex road networks with many junctions and roads with various traffic signalization and regime. In addition, road network editor is implemented in order to allow easy and user friendly road network reconstruction.

Keywords—Traffic simulator; MovSim; microscopic traffic model; road network editor.

I. INTRODUCTION

The reason for growing interest in traffic simulation models lies not only in their potential for modelling traffic processes, but also their ability to visualize the results in a realistic and appealing way [1].

Traffic network simulation is an important approach for modelling the dynamics of the traffic systems [2]. When implemented with enough details of the real network it helps analyzing the locations and causes of congestions. In this manner, it can offer solutions for uninterrupted traffic flows and traffic safety. The way of traffic congestion evaluation significantly affects transport-planning decisions [3].

Traffic jams occur at any hour, either weekdays or weekends. Drivers face extra travel time, extra cost from wasted fuel, lost productivity as well as increasing unreliability where bad weather, roadwork, a malfunctioning traffic signal, a local event, a small accident, or stalled vehicle can result in major delays [4]. In the 2014 Annual Report, INRIX reveals that congestion levels rose in over half (53%) of European cities.

The microscopic-based simulation provides a detailed representation of the traffic process, being suitable for evaluation of complicated traffic facilities and intelligent transportation systems that often consist of complex traffic management, safety and information systems [1]. However, mesoscopic and macroscopic simulation represent the traffic dynamics with less details, but are faster and easier to apply and calibrate. While macroscopic approach is more suitable for modelling large networks, microscopic approach is usually applied to smaller areas.

To allow detailed and accurate traffic simulation, including congestion evaluation, the road network needs to be described with sufficient details. These include sections lengths, number of lanes, curvature, lane restrictions, etc. All details have a great influence on simulation results. In this paper, we present

the upgraded Multi-model open-source vehicular-traffic Simulator, MovSim, which supports complex road network construction. The implemented GUI allows easy road network adaptation to the appearing network topology changes. Even when all details from the real-world are captured in a simulation model, the upgrades and updates of the road network characteristics are still allowed. This includes for instance the possibilities to easily close/open some road section, to easily change the traffic light control sequence, as well to easily adapt the topology of the complex road crossing.

The rest of this paper is organized as follows: Section II lists some related work on traffic simulations; Section III briefly presents the simulator that was used as an origin for our upgrades; the upgrades are described in Section IV; road network editor is described in Section V; while Section VI draws the concluding remarks.

II. RELATED WORK

Traffic simulation models can be classified into [5]:

- microscopic - approaching individual elements of transportation systems, such as individual vehicle dynamics and individual traveler behavior;
- mesoscopic - approaching transportation elements in small groups, within which elements are considered homogeneous;
- macroscopic - approaching aggregated characteristics of transportation elements, such as aggregated traffic flow dynamics.

MovSim [6] is an interactive Java-based open-source traffic simulator. In contrast to commercial simulators, the focus is on investigating fundamental issues of traffic dynamics rather than simulating specific road networks. Multi-agent Transport Simulation, MATSim [7] provides a framework to implement agent-based transport simulations. The framework consists of several modules which can be combined or used stand-alone. Modules can be replaced by user's own implementations. Simulation of Urban MObility, SUMO [8] is an open source, microscopic and continuous road traffic simulation package designed to handle large road networks. It allows modelling of intermodal traffic systems including road vehicles, public transport and pedestrians. It can be enhanced with custom models and provides various APIs to remotely control the simulation.

On the mesoscopic and macroscopic level of traffic simulation the following models are available. Mesoscopic Traffic Simulator, Mezzo [9] is a discrete-event mesoscopic traffic

simulation model that simulates road traffic on the level of individual vehicles, but with an aggregated behaviour on links. The model is especially designed to simulate large networks and can be used in combination with a microscopic simulator to work as a hybrid model. OmniTRANS [10] is a macroscopic multimodal and multitemporal system, suitable for modelling the interactions between the various means of transport within an urban context. PTV Visum [11] is a macroscopic traffic analysis and forecast tool that provides a comprehensive range of functions for all aspects of transportation planning and engineering.

III. MOVSIM TRAFFIC SIMULATOR

The aim of vehicular-traffic simulator MovSim [6] is to be an open-source tool for model investigation and simulation. It can be classified as a microscopic lane-based traffic simulator. It is able to simulate many basic traffic situations. Simulated vehicles are able to react to a traffic light, and are able to overtake. Lane changes are modeled with the general (minimizing overall braking induced by lane change, MOBIL) strategy based on accelerations [12]. Simulator is supported with graphical user interface, where simulated vehicles dynamics is visualized. It is written in Java and Maven is used as management tool.

The simulator offers the implementation of vehicles inflows and outflows. Inflows are points where vehicles enter the simulated traffic network. They are defined by the network position and the numbers of vehicles entering the network per second. Outflows are points where vehicles exit simulated network. The construction of simple single lane road segment with inflow and outflow, where vehicles are moving from left to right with a setting of input flow to 1000 vehicles per hour is shown in Figure 1.



Figure 1. Single-lane road with vehicles flow.

For every vehicle a lane-changing politeness parameter is assigned, which vary the motivation for lane changing from egoistic to more cooperative driving behavior. In this respect solutions used in MovSim, such as the lane-changing MOBIL passing rules and longitudinal car-following model Intelligent Driver Model (IDM) [13] are essential for simulation of multilane traffic.

A code describing the following traffic cases was available:

- roundabout;
- road block;
- inclusion road in traffic;
- speed limits;
- behavior of vehicles driving on ramp.

In order to built real complex traffic network with many junctions and roads with multiple lanes and various traffic regimes, some upgrades of MovSim were needed. In this paper, upgraded simulator with an editor for easy manual road reconstruction of existing network is presented.

IV. SIMULATOR UPGRADE

A. Simple junction and lanes with multiple sources and sinks

In order to support the simulation of more complex road structures, such as junction the simulator was upgraded with functionality which allows a leap from one to several road segments. This was required to build even a basic single-lane junction. A basic single-lane junction consists of four input and four output road segments, each having one lane. In basic single-lane junction four roads intersect in a way such that the change from one road to another is allowed.

Upon entering a basic single-lane junction there are three options, the vehicle can turn left, right, or go straight. Similarly, vehicle which leaves the junction could come from three directions. The original MovSim supports only one source and one sink lane segment for each lane. Therefore, the simulator was upgraded with necessary changes so that each lane can have more sources and sinks. Further, in the markup language we had to define a junction. A basic junction consists of four input road segments, four output road segments, and 12 road segments that connect input and output road segments. For an illustration see Figure 2.



Figure 2. Basic junction.

Then, basic junctions were upgraded by adding traffic lights and setting their lights duration. All four traffic lights are automatically synchronized only by setting the duration of green, yellow, and red of one traffic light direction.

By the simulator upgrade, which allows assigning several sinks to one lane in the road segment, the vehicle route simulation is not trivial anymore. For example, when vehicle comes to the junction, it has no information, which road it should take (left, right, or straight in the basic junction). Therefore, to prevent errors, a sequence of consecutive road segments (path) needs to be defined in advance for every vehicle. In practice, all possible routes from a given source to a given sink are calculated. When a vehicle is generated, one of the possible routes is chosen at random and assigned to it. Based on the roads and their connections, a graph is generated. Its vertices represent links between road segments (sources and sinks of each segment) and edges represent road segments. A graph was build using Java library JGraphT [14]. In order to determine most favorable routes between source and sink (origin and destination) Dijkstra algorithm [15] was used.

B. Critical parts of roads

It seems to be impossible to avoid intersection of road segments in the complex road networks construction. Even in

road network with basic junction such situation appears. A nice example of critical road segments intersection in basic junction is when cars are turning left. A car cannot turn left until the road to be crossed is empty. In order to prevent car crashes, two critical segments, named $road_1$ and $road_2$, of intersecting roads are defined:

- $road_1$ represents problematic part, in which road segments intersect;
- blockade on $road_2$: blockade is set, while $road_1$ is busy (e.g., some vehicle is driving on it).

Traffic through critical parts are also adjusted by defining acceptable speed of vehicles on the critical parts of the road. If the approaching vehicle drives very slowly through the $road_1$, then the blockade on the $road_2$ for turning left is not needed since the turning vehicle has enough time to turn left.

C. Road network construction

Our testing case of real-world road network construction was Ljubljana road network within the Ljubljana ring. MovSim simulator does not have any tool for easy real-world road network construction. Basic small road network construction was done with the help of image of a city map, which was set as the background in the canvas of graphical user interface. City map was not enough to construct the road network, since on the map, it was not evident how many lanes are in input road segments of a junction. Such vagueness was manually solved with the help of Google street-view service. The construction of simple basic junctions were automatized to some level, while the construction of complex junctions is facilitated but it still needs to be done manually.

An example of the detailed construction of small subnetwork consisting of three complex junctions, where Tivolska, Celovška and Slovenska streets of Ljubljana city intersects, can be seen in the Figure 3. Each road segment is identified by unique ID number. Each junction has input roads, output roads, roads within the junction as well there are roads that connect junctions.



Figure 3. Partial road network.

In the complex network (roads with multiple lanes and multiple output roads), the following problems were detected.

Increased number of vehicle crashes. When larger number of vehicles entered the network, they started to crash. The reason is that the relaxation of vehicles is computationally time consuming. Relaxation is calculation of vehicle priorities order. Most of the collisions occurred during the transition of vehicles from different road segments that have the same sink/target segment. For example, when two vehicles from different road segments try to enter the same road segment at the same time, the calculated positions are the same and the collision is recognized. The problem was solved with similar observers, such as critical parts of the road segments, which were used for simple basic junctions, when vehicles are turning left.

Some vehicles cannot continue their routes. As already mentioned, the route from the source to the sink is assigned to the vehicle before it enters the network. It turns out that some vehicles cannot continue its route. The most common reason is an inability to change the lane. If the vehicle cannot change the lane, it continues its ride to the segment defined as a successor of the current lane. Such vehicle is lost since its road position deviates from originally assigned route. A simple solution can be to simply exclude lost vehicles from the traffic. To make the simulation closer to the real-world situation, the problem was solved by the route recalculation; i.e., a new route from the current road position to the sink is assigned.

D. A network of several roads within the Ljubljana ring

We constructed a network consisting of Ljubljana ring, main basic junctions inside the ring, and roads connecting them (see Figure 4). During this construction the necessity of automatization and facilitation of such network construction arose. Junctions were positioned by mouse clicks and a code of all road segments for all junctions was generated in XML markup language. The result was a set of junctions with missing interconnections. To achieve correct interconnections road identification numbers were needed. Hence, the road network editor for easier and quicker road network composition was required.



Figure 4. Roads within Ljubljana ring.

We have also started much more demanding part of the road network editor upgrade, i.e., building a parser for automatic road network construction based on OpenStreetMap [16]. The first thing to do was detecting where roads intersect. When positions of all junctions were calculated, it was necessary to get details of each junction from geometry of roads that intersect in that particular junction. Simple directed graph turned out to be very useful for both storing data and converting it to roads and junctions.

V. ROAD NETWORK EDITOR

We built a graphical user interface in order to facilitate the composition of road networks and to avoid manual entering of every individual road segment into the OpenDRIVE [17] file.

Road network editor supports different view options. During the traffic simulation, user can choose the content of the canvas. Possible elements are: map of the city, road segments, traffic lights, ids of roads, sources of vehicles, sinks of vehicles, and speed limits, see Figure 5.

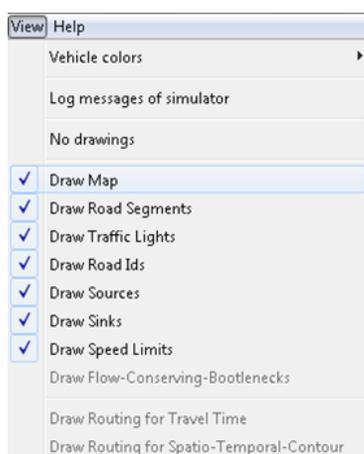


Figure 5. View options.

The most important part of the upgraded GUI is an Edit option which is added to the GUI's toolbar. By selecting edit mode, the simulation stops and the editor is prepared for applying network reconstruction. In edit mode, new road segments and junctions can be created as well as new traffic lights with their light intervals setting can be added. In the edit mode a virtual wrapper around segments is created for easier segment identification and selection.

If straight road is added, the information about x and y coordinates, segment length, and angle under which the road lies are shown in the toolbar. If the curved road is added, also radius and arc length are shown in the toolbar. By pressing the Apply button, the road is added to an existing network but it is not stored in XML structure yet. If we want the road to be stored in the network, we need to explicitly save changes.

Road editor supports adding interconnections of road segments, which are essential for vehicle recognition of the following road segment on the route (see Figure 6). By adding the interconnection of two roads, the lane which represents successor (sink) and lane which represents predecessor (source) must be defined.

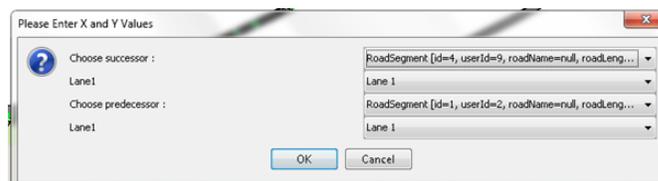


Figure 6. Connection dialog box.

A. Adding junctions

The first problem in a junction construction was to decide which properties of junctions were sufficient for reconstruction. It turned out that there are many properties that need to be determined in order to define junction geometry. These needed properties are evident from the later explanation of junction editor. Further, it is not easy to build a junction out of numbers describing it using just simple geometric shapes as lines and circle arcs. Finally, roads going through the junction need to be linked so that the constructed junction could be used in simulation.

An option for adding simple junction was added in the road editor. By choosing junction scope, Junction option appears. By clicking it (see Figure 7), a Junction editor window opens

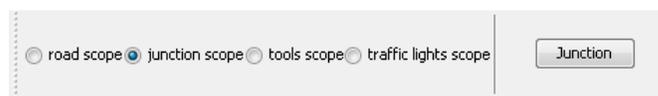


Figure 7. Editing toolbar.

(see Figure 8). The user can choose among several junction types, e.g., T, Y, or $+$. Further, the user needs to define width of roads within the junction, lengths of external roads (incoming and outgoing roads), (x, y) position, rotation angle of the junction, and the number of incoming and outgoing roads for every leg of the junction. For example, if the junction of $+$ type is chosen, then there are 4 legs, and for each of them the number of incoming and outgoing roads must be defined.

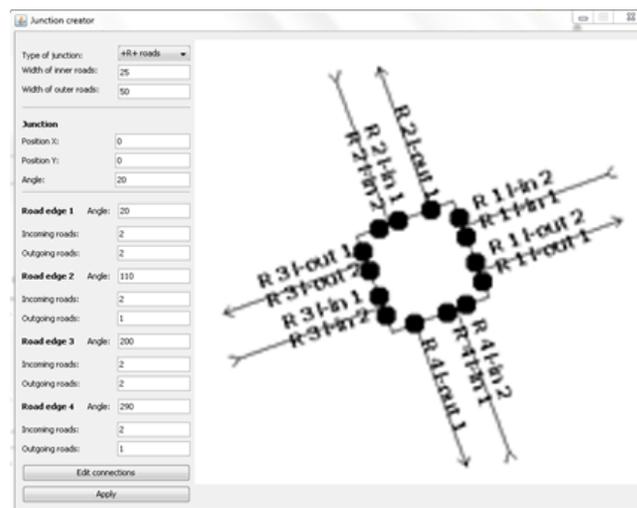


Figure 8. Junction editor.

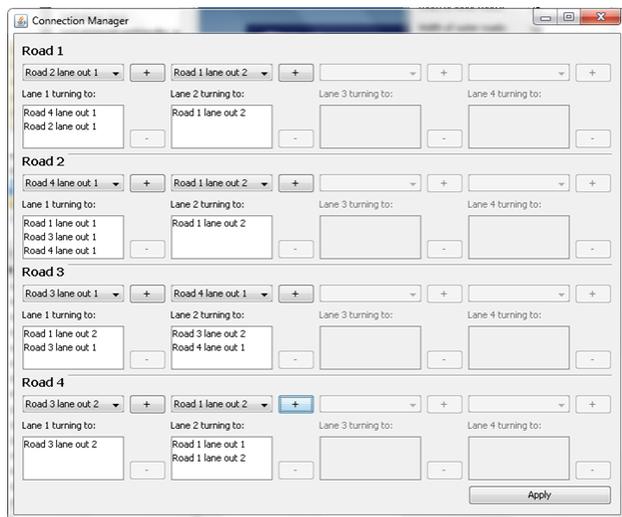


Figure 9. Editing toolbar.

Note that the above-mentioned information is not enough for the junction definition. This information would represent only a set of roads with some dimensional properties. To make it complete, individual roads inside the junction must be connected. Once we fulfill all the information that we want for the junction, an Edit Connections option appears. By clicking it, Connection Manager window opens (see Figure 9). This editor allows connecting incoming and outgoing roads. Each edge of the junction can have up to four incoming roads. For each incoming road the outgoing roads with which it is associated are listed.

B. Traffic lights editor

Traffic lights are mostly assigned to the lanes on the roads in junctions. Different traffic light intervals/phases and their duration in the junction are mutually dependent and are uniquely determined by assigning adequate Controller Group. Although many junctions may have the same traffic light regime, their phases do not always overlap. Mutual dependence of periodic repetition of different light phases in traffic lights in different junctions are controlled by assigning starting time in each junction.

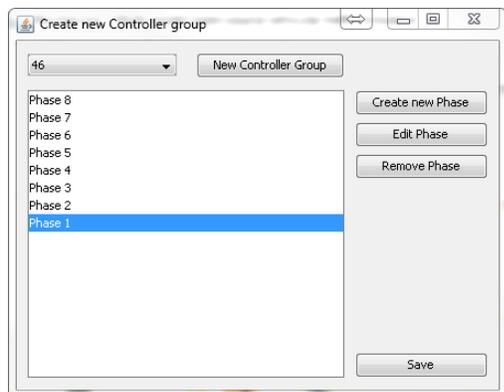


Figure 10. Controller Group editor.

Editing of traffic lights consists of Controller Group part

and traffic light location part. First, in the Controller Group part, the light regime for a given set of traffic lights is defined (see Figure 10). The set of traffic lights typically refers to the traffic lights of the junction. Second, the location of traffic lights is determined. These two options can be found as two buttons under traffic lights scope option. The buttons are named Create Group Controller and Locate Traffic Light. In Create Group Controller editor controller groups of traffic lights can be managed. New controller groups can be optionally added, as long as we need them, or they can be selected among the existing ones. Each Controller Group has phases, where each phase is defined by its duration and color of traffic lights. Phases can be added, edited, or removed. Phases are shown in list where each of them can be selected and edited. If we want to edit selected phase, we click Edit

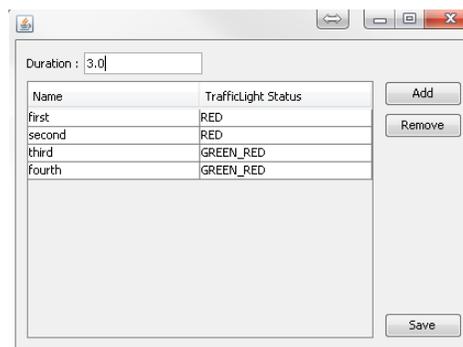


Figure 11. Phases editor.

Phase. In the Phase Editor, we have to specify the length of the phase (see Figure 11). Traffic lights can be optionally added and removed. When the editing of the phase and controller group is finished and saved the data is written to the XML OpenDRIVE structure.

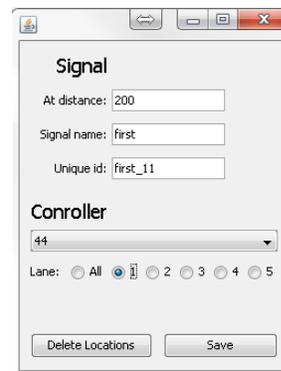


Figure 12. Traffic light locator.

In order to locate traffic lights on the road, the road segment must be chosen. By choosing the road segment, Locate Traffic Light editor opens (see Figure 12). In the editor, the distance from the ending point of the road is specified. Then, the same name as it was set in phases in Controller Group must be entered. Further, an unique id number by which the traffic light is recognized by others is assigned. Finally, the Controller Group to which it belongs to and the lane on which it works is selected.

VI. CONCLUSION

The focus of this paper was given to the development and implementation of microscopic traffic simulation, which can be helpful at analysis of existing traffic congestions in different city parts. Such microscopic traffic simulation can offer a solution for uninterrupted traffic flows and traffic safety. Since a comprehensive open-source traffic simulator MovSim already supports many options for model investigation and simulation, we upgraded it so that it supports the construction of complex road networks with many junctions and road segments with multiple predecessor and successor road segments. In addition, we built an user-friendly tool for a real-world road network construction in order to facilitate the construction of real-world road networks and to avoid manual entering of individual road network parts into the XML OpenDRIVE file. GUI supports automatic construction of simple junctions with all internal road segments and traffic lights with an option of light phases adjustment.

In the future, we plan to built the parser for automatic road network construction based on the OpenStreetMap data where information will be automatically considered and included.

ACKNOWLEDGMENT

This work was partly funded by the Slovenian Research Agency through the research program "Computer Structures and Systems", and was also supported through ARTEMIS-JU funding for the project "ACCUS – Adaptive Cooperative Control in Urban (sub) Systems" (grant number 333020). [?]

REFERENCES

- [1] W. Burghout, "Hybrid microscopic-mesoscopic traffic simulation", Doctoral Dissertation, Royal Institute of Technology, Stockholm, Sweden, 2004.
- [2] M. Treiber and A. Kesting, "Traffic Flow Dynamics - Data, Models and Simulation", Heidelberg, Springer, Oct. 2013.
- [3] T. Litman, "Comprehensive Evaluation of Traffic Congestion Costs and Congestion Reduction Strategies", August 2015, URL: http://www.vtpi.org/cong_relief.pdf, [accesses: 2015-09-02].
- [4] "INRIX, Urban Mobility Scorecard Annual Report", URL: <http://inrix.com/scorecard/key-findings-us/>, [accesses: 2015-09-02].
- [5] R. Akçelik, "Traffic models", URL: <http://www.sidrasolutions.com/Resources/Models>, [accesses: 2015-06-03].
- [6] M. Treiber and A. Kesting, "An Open-Source Microscopic Traffic Simulator", *Intelligent Transportation Systems Magazine*, vol. 2, no. 3, pp. 6–13, 2010.
- [7] "MATSim", URL: <http://www.matsim.org/>, [accesses: 2015-06-03].
- [8] "SUMO - Simulation of Urban MObility", URL: http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/, [accesses: 2015-06-03].
- [9] "Mezzo - Mesoscopic Traffic Simulator", URL: <http://www.ctr.kth.se/mezzo.php>, [accesses: 2015-06-03].
- [10] "OmniTRANS", URL: <http://www.dat.nl/en/products/omnitrans/>, [accesses: 2015-06-03].
- [11] "PTV Visum", URL: <http://vision-traffic.ptvgroup.com/en-us/products/ptv-visum/>, [accesses: 2015-06-03].
- [12] A. Kesting, M. Treiber, and D. Helbing, "General Lane-Changing Model MOBIL for Car-Following Models", *Journal of the Transportation Research Board*, no. 1999, pp. 86–94, 2007.
- [13] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations", *Physical Review E*, vol. 62, no. 2, pp. 1805–1824, 2000.
- [14] "JGraphT", URL: <http://jgrapht.org/>, [accesses: 2015-07-01].
- [15] E. W. Dijkstra, "A note on two problems in connection with graphs", *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
- [16] "OpenStreetMap", URL: <https://www.openstreetmap.org/>, [accesses: 2015-07-01].
- [17] "OpenDRIVE", URL: <http://www.opendrive.org/>, [accesses: 2015-07-01].

Evaluation Method for Appropriate Fleet Size of Forklifts at a Freight Station

Shunichi Tanaka

Signalling & Transport Information Technology Division
 Railway Technical Research Institute
 Kokubunji Japan
 tanaka.shunichi.18@rtri.or.jp

Abstract— In a freight station in Japan, the number of forklifts is determined experientially based on the quantity of work, shape of a container platform and operating time. But, it is difficult to evaluate the appropriateness of the number of forklifts quantitatively. So, the author developed a simulator of cargo handling operations in freight stations, and studied influences of the fleet size on the works of loading and unloading. As a result, it was found that the waiting time of trucks for loading or unloading was correlated with operation rate of forklifts. The shortening of the waiting time of trucks is desirable from the point of view of customer satisfaction. In this paper, the author proposes a method to evaluate the appropriateness of the size of the forklift based on operation rate of forklifts.

Keywords- Fleet size of Forklifts; Freight Station; Cargo handling operation

I. INTRODUCTION

Rail freight transportation in Japan is mostly operated by Japan Freight Railway Company (JR Freight). Most of the lines that JR Freight uses are owned by passenger railway companies, and JR Freight operates their trains by making them wedged in between the operation of passenger’s trains.

Japanese passenger railway is well known for its high density and punctuality, so that freight trains also must be punctual for arriving or departing time.

Over 700 freight trains are operated on the average each day in Japan. Of these, about 400 trains are container trains [1]. The freight stations used to handle container trains have “container platforms” surrounding the tracks for loading and unloading containers, and containers are loaded onto trains from container platforms or unloaded from trains onto container platforms by forklifts(Fig. 1). Transportation companies that are commissioned to take care of the containers go to container platform and load or unload containers onto/from own truck there using forklifts.

As large-scale freight station must handle over 50 trains (sum of arrival and departure trains) per a day, schedule of freight cars and locomotives in such large-scale freight stations must be planned in detail. In addition, most transportation companies bring out their containers in a few hours from arrival of freight trains, and some companies bring in their containers just before a deadline. Therefore, handling operation becomes very complicated and difficult.

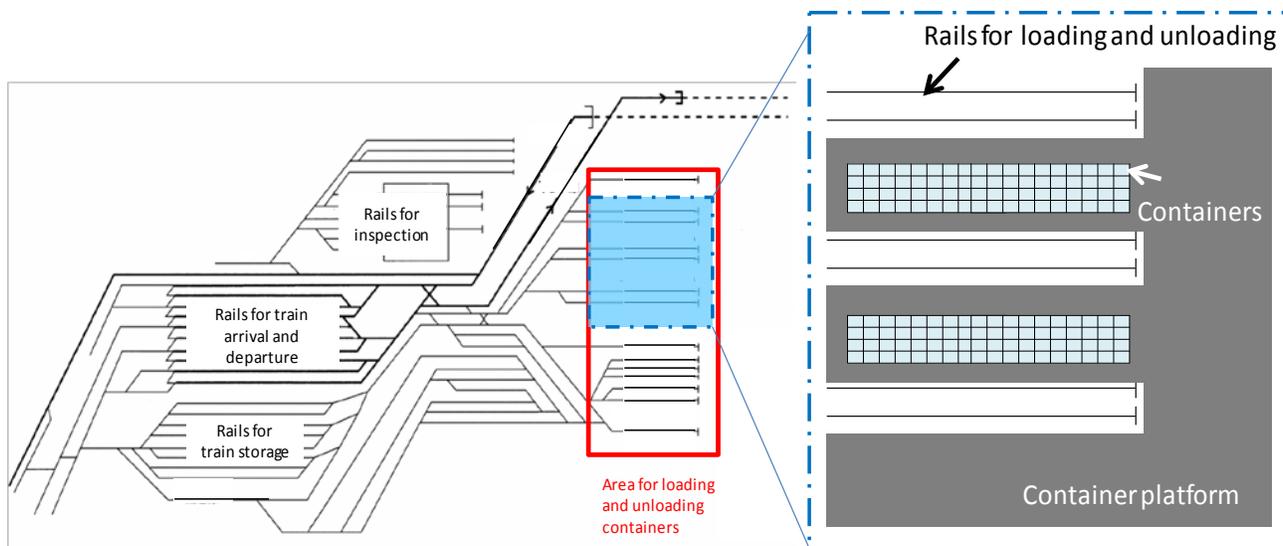


Figure 1. Example of Japanese Freight Station

Figure 2. Classification of cargo handling operations

TABLE 1. PRIMARY OPERATIONS

Brought in by	Brought out by	Container handling
Storage	Storage	No handling
	Train	Storage → Yard truck → Container platform → Freight car
		Storage → Freight car
Forwarder	Storage → Forwarder’s truck	
Train	Storage	Freight car → Container platform
	Train	Freight car → Yard truck → Container platform → Freight car
		Freight car → Container platform → Freight car
Forwarder	Freight car → Container platform → Forwarder’s truck	
Forwarder	Storage	Forwarder’s truck → Container platform
	Train	Forwarder’s truck → Container platform → Freight car

At Japanese freight stations, forklifts and other cargo handling equipment are used to load and unload containers. Up until now, decisions on how many forklifts should be assigned to a freight station have been based on experience, based on a number of cargo works, a shape of the yard, working hours, and so on. It is difficult to determine whether the current number of forklifts assigned is too large, too small or appropriate. Quantitative analysis for appropriate assignment of resources has been a challenge. Against such background, the author developed a simulator program that is capable of analyzing the impact of the number of forklifts assigned to a freight station against the loading operations at the station. This paper introduces how the simulator program is used to assess the adequacy of the number of assigned forklifts.

JR Freight has introduced the system “IT-FRENS SYSTEM (IT FREight information Network System)” that integrates management of containers in 2004 [2]. By using this system, we can grasp move of containers and move of forklifts in a freight station. The author developed the cargo handling simulator based on analysis of these data.

The behavior of forklifts has been studied in the domain of warehouse. These researches aim at creating efficiency of picking or optimizing layout of a warehouse [3][4][5]. Therefore, it is difficult to apply the results of these researches to Japanese freight stations. The loading in ship transportation has been studied for many years [6][7][8], but it is difficult to apply these study results to the railway sector because the feature of loading and unloading facilities, and the method of loading and unloading in ship transportation are different from these in railway transportation.

II. CLASSIFICATION OF CARGO HANDLING OPERATIONS AT FREIGHT STATIONS

The author classified the cargo handling operations at freight stations using the operation histories of forklifts used

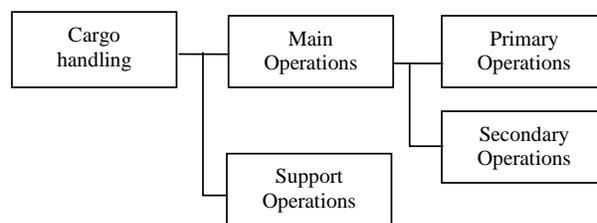


Figure 2. Classification of cargo handling operations

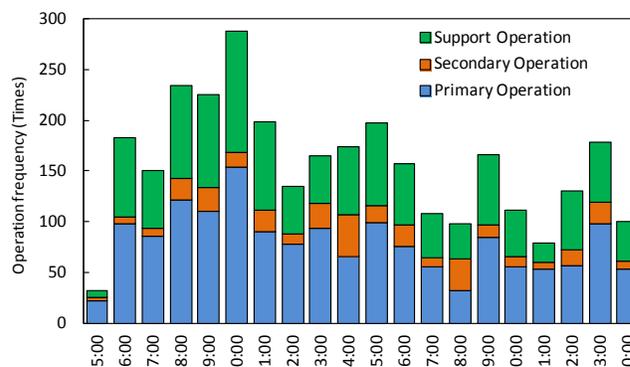


Figure 3. Frequencies of Support, Primary and Secondary Operations

at the stations. As shown in Fig. 2, cargo handling operations are classified into Main Operations (moving the container) and Support Operations (moving away containers above or in front of a container to gain access to the container). Furthermore, Main Operations are subdivided into Primary Operations (operations typically carried out from when the container is brought into the station up to when the container is brought out, which are outlined in Table 1) and Secondary Operations (rearranging the station premises and cargo stacks to facilitate the handling of the container). Frequencies of Support, Primary and Secondary Operations at a freight station are shown in Fig. 3. It indicates that Support Operations account for roughly 40%

of all operations and that there are only small variations in the ratio between Support and Primary Operations throughout the day.

III. LIMITATIONS ON CARGO HANDLING

Typical limitations on cargo handling are described below.

A. Time-related limitations

- The time specified in the yard operation plans (i.e., time for trains pulling into and out of the container platform) must be observed.
- Waiting time of the forwarders' trucks should be minimized to achieve customer satisfaction.
- Appropriate sequence of cargo handling shall be taken into consideration, such as the flow of Primary Operations shown in Table 1, loading and unloading operations performed in the same area, etc.

B. Limitations posed by variations of containers

- Some sizes of cargo handling machines must be used in accordance with the size of container to be handled.
- At most freight stations, storage areas of the containers are specified according to their sizes

C. Spatial limitations

- Support Operations need to be considered according to the current container storage status.
- If the specified container storage space is fully occupied, excess containers shall be stored in nearby vacant space.
- The container platform has only a limited space for moving and some areas have restrictions such as

one-way traffic.

- The cargo handling operations should be arranged so that the total distance travelled by forklifts is kept as short as possible.

To estimate the appropriate number of forklifts that should be assigned to a freight station to properly perform cargo handling operations under the limitations described above, a clear picture of the operations of the forklifts in a given environment including the layout of the station, the status of the containers and temporal distribution of the cargo handling work needs to be formed. For this purpose, the author developed a simulator program.

IV. DEVELOPMENT OF CARGO HANDLING SIMULATOR PROGRAM

A. Modeling of freight station

Container platforms are normally complex in shape as they are constructed around cargo handling tracks, which makes simulation difficult. Moreover, as the locations of container yards and the areas where forklifts can travel are restricted, the distance a forklift actually travels is not the direct distance between any two points. To overcome this, the container storage areas at a freight station were created in a multidimensional space by defining the time taken to travel between two storage areas as a corresponding distance in a Euclidean space.

For each area, arrays for the number of containers that can be stored on that container platform, trucks and freight cars (if the area is served by adjoining tracks) were assigned.

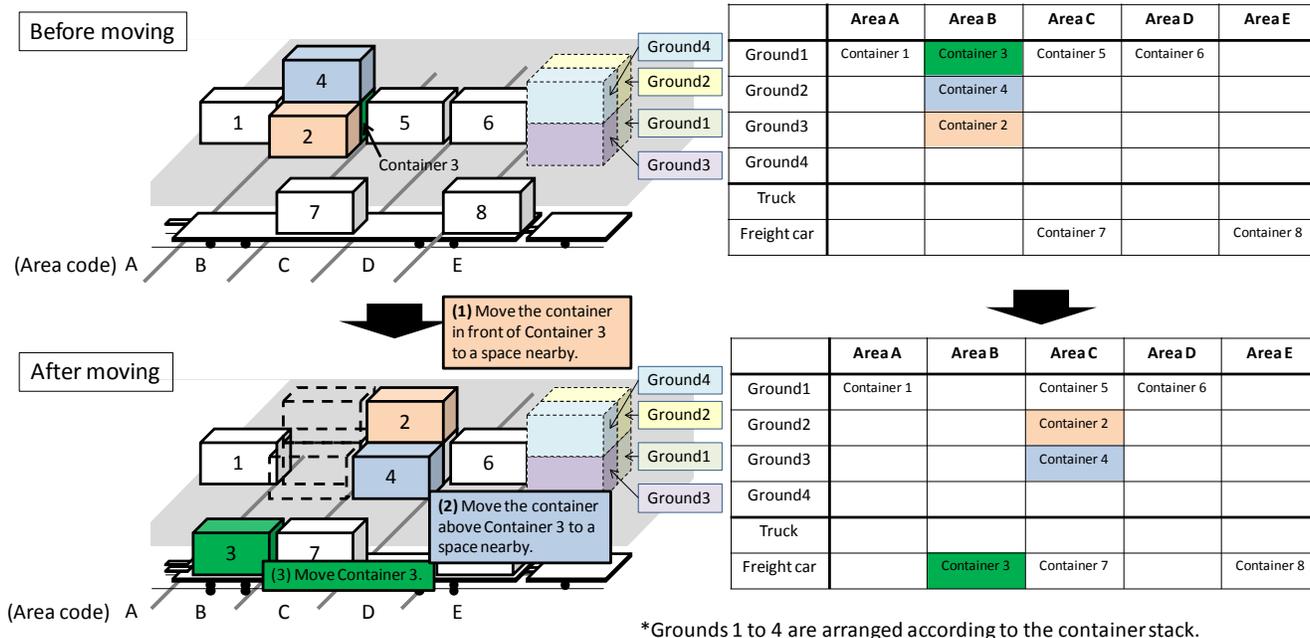


Figure 4. Modeling of storage area

The storage conditions of that area were explained by storing the container IDs in the arrays. The algorithm was designed in such a way that containers that were behind or beneath other containers could not be moved, and so that container IDs could not be stored in the train or truck array unless trains or trucks were available. For the latter, operation management data of trucks and trains is utilized.

The modeling enables changes of container locations made by forklifts to be represented by changes of the relevant container IDs stored in the arrays. For example, as shown in Fig. 4, moving Container 3 to Area B of a freight car requires a Support Operation consisting of moving away Container 2 from in front of Container 3 as well as Container 4 from above Container 3. These location changes are represented by the relevant changes in the tables in Fig. 4.

B. Simulation algorithm

The developed simulator program employs a time-driven

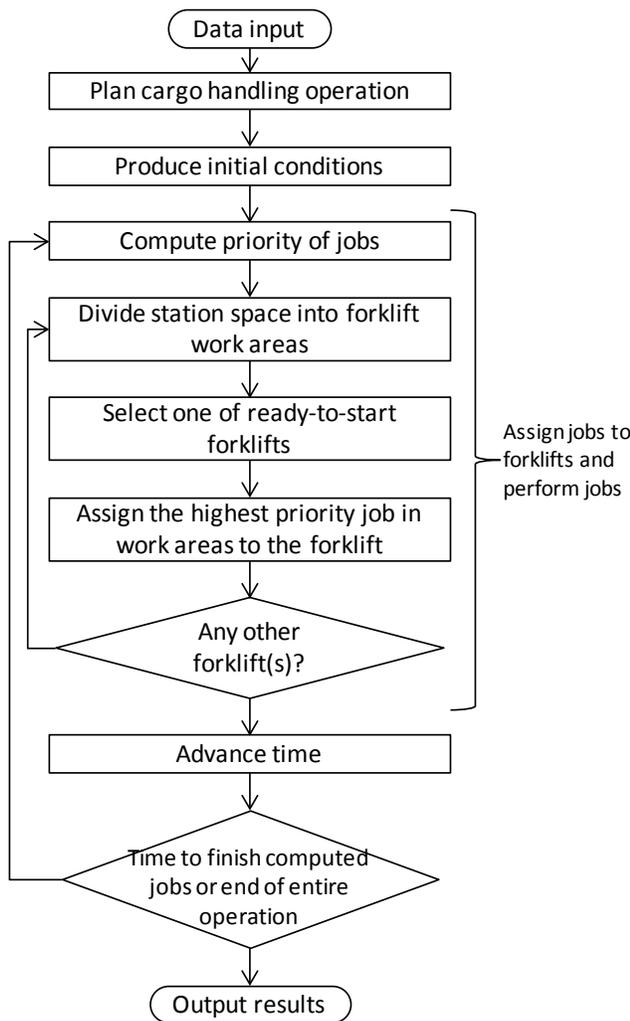


Figure 5. Simulation's computational flow

algorithm for assignment of cargo handling jobs. The algorithm is shown in Fig. 5. The simulation utilizes not just the arrays of containers stored on the station premises but a database on the status of available forklifts and another database on the status of Main Operations. Changes on the station premises are produced by changing these data with a lapse of time. The Main Operation database is created based on input data. Each Main Operation job is given a ready-to-start time. The arrival of a train or truck can be simulated by assigning the ready-to-start time to the time when the train or truck has arrived and cargo handling operation has become possible. Each record in the Main Operation database is time-sequence enabled to ensure a job (e.g., loading a container to a truck) cannot be started until the previous job (e.g., unloading an existing container from the truck) is completed. This can also be applied to Primary Operations in Table 1.

Job assignment to available forklifts is determined based on a prioritized order of ready-to-start jobs, which is calculated based on job types, deadlines, wait time and other parameters, and the work areas of forklifts, which is calculated based on the distribution of cargo handling jobs and forklifts and the operation status of forklifts. The method of determine the priority order depends on custom and circumstance of each station so that we must prepare an expression corresponding to each station.

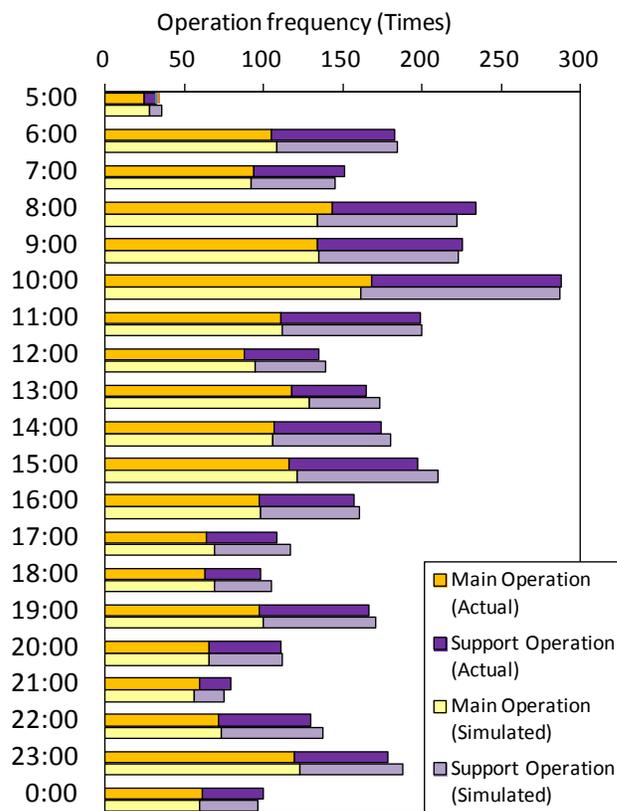


Figure 6. Comparison of actual and simulated operation frequencies

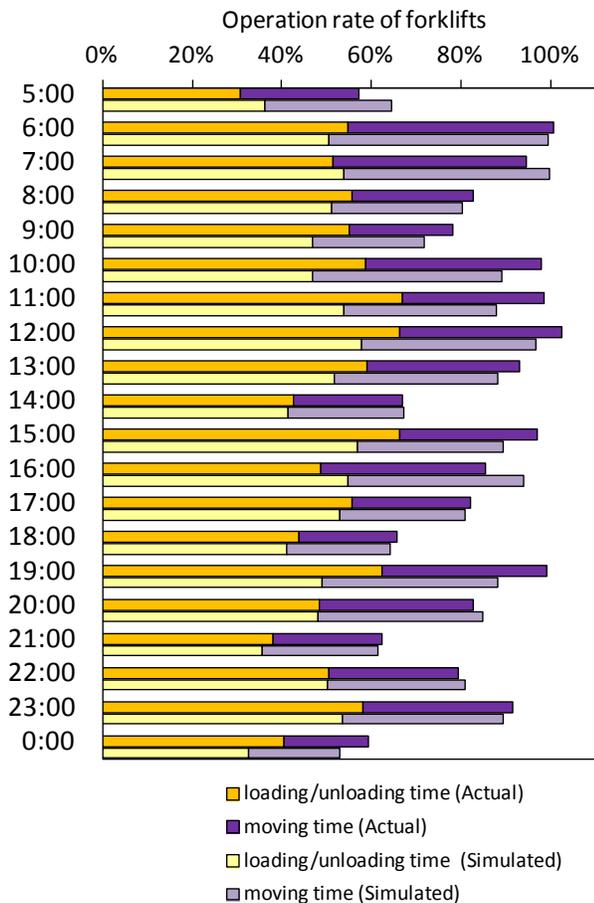


Figure 7. Comparison of the ratio of moving time and loading/unloading time

C. Verification of cargo handling simulator program

The simulation program was verified using the actual data of a freight station. Data covering one day at the station was imported and a simulation was conducted for comparison between the actual data and the simulation result. Fig. 6 shows the frequencies of Main and Support Operations for each time zone. Fig. 7 shows the ratio of time spent moving containers out of the forklift operation time for both Main and Support Operations. Both figures indicate a good reproduction of the actual data. With the results, the simulator can be considered accurate enough for practical application to forklift fleet size evaluation.

V. FLEET SIZE EVALUATION METHODS USING CARGO HANDLING SIMULATOR PROGRAM

The simulator program offers the facility to increase and decrease a forklift fleet size and observe the results. Through a trial and error process with the simulator, it is possible to determine the size of a forklift fleet that offers efficient operation with minimum idle time. The result can then be used to evaluate the current fleet size. However, performing

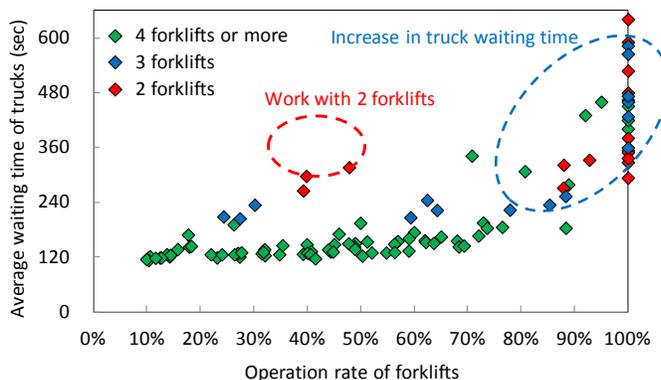


Figure 8. Relationship between the operating rate of forklifts and the average waiting time of trucks

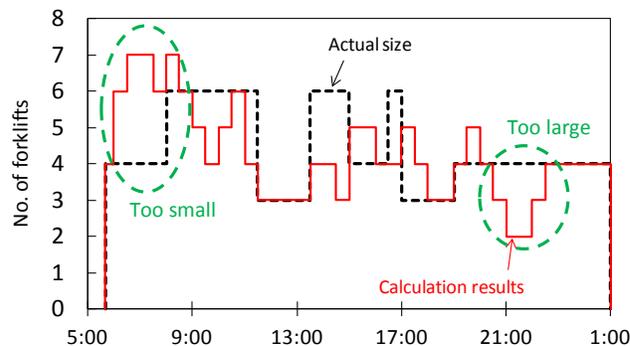


Figure 9. Calculation of efficient fleet size

such evaluation while changing the fleet size with hours can involve too many combinations to consider. Instead, simulation was performed to find out how average waiting time of forwarders' trucks could be minimized.

For this simulation, using actual facility data of a freight station, cargo handling jobs were randomly generated while the size of a forklift fleet was changed. The relationship between the operation rate of forklifts and the average waiting time of forwarders' trucks is shown in Fig. 8.

With a fleet of two forklifts, the average waiting time can be relatively long even when the operating rate of the forklifts is low, indicating the fleet is too small for the size of the station (i.e., too much time spent traveling between containers on the premises). With a fleet of three forklifts or more, the average waiting time starts to increase suddenly when the operating rate becomes around 80 to 90%. In other words, keeping the operating rate below 80% can be effective in preventing the waiting time from becoming too long.

Keeping the 80% threshold in mind, it is possible to identify the time when the fleet size is too small or too large by calculating the fleet size that can keep the operating rate below 80% based on actual workload data and then comparing the result with the actual fleet size. (Fig. 9 shows

the result of calculation, in which the fleet size was changeable every 30 minutes.) By observing the actual data for several days to check if chronic fleet excess or shortage occurs at any time of the day, the appropriateness of the fleet size can be evaluated.

VI. CONCLUSION

The author developed a simulator program to simulate the cargo handling operations at a freight station and proposed a method to evaluate the appropriateness of the size of the forklift fleet using this program. In determining the actual fleet size for an existing freight station, however, a range of parameters still need to be considered including operators' work schedules and variations in cargo handling workload.

ACKNOWLEDGMENT

The author is grateful to Japan Freight Railway Company for their cooperation.

REFERENCES

- [1] K. Iwata, "Rail Freight in Japan –The situation Today and Challenges for Tomorrow" , Japan Railway & Transport Review 26, February 2011, p.p. 8-17.
- [2] T. Hanaoka, ""Values" created by IT-FRENS & TRACE systems", Japanese Railway Engineering, Vol.51,January 2011, p.p.1-5.
- [3] S. Takakura, H. Takizawa, and K. Ito, "Simulation and analysis of non-automated distribution warehouses", Proceedings of the 32nd conference on Winter simulation, 2000, p.p.1177-1184.
- [4] Gagliardi, J. P., Renaud, J., and Ruiz, A, "A Simulation Model to Improve Warehouse Operations", Proceedings of the 2007 Winter Simulation Conference, 2007.
- [5] C. Y. Liong and C. S. E. Loo, "A simulation study of warehouse loading and unloading systems using ARENA", Journal of Quality Measurement and Analysis, 2011, p.p. 45-56.
- [6] M. Avriel and M. Penn, "Exact and approximate solutions of the container ship stowage problem" , Computers & Industrial Engineering Volume 25, Issues 1–4, September 1993, p.p.271-274.
- [7] I. D. Wilson and P. A. Roach, "Exact and approximate solutions of the container ship stowage problem" , Journal of Heuristics , Volume 5, Issue 4, December 1999, p.p. 403-418.
- [8] A. Moura, J. Oliveira, and C. Pimentel, "A Mathematical Model for the Container Stowage and Ship Routing Problem", Journal of Mathematical Modelling and Algorithms in Operations Research, Volume 12, Issue 3, September 2013, p.p. 217-231.

Statistical Emulation Applied to a Very Large Data Set Generated by an Agent-based Model

Wim De Mulder
and Geert Molenberghs
and Geert Verbeke

Leuven Biostatistics and
Statistical Bioinformatics Centre
Leuven, Belgium
Email: wim.demulder@cs.kuleuven.be

Bernhard Rengs
and Thomas Fent

Wittgenstein Centre (IIASA, VID/ÖAW, WU)
VID/ÖAW
Vienna, Austria

Abstract—This paper presents a method to apply statistical emulation on very large data sets, making use of cluster analysis. It is shown how integrating cluster analysis with the interpolation method called inverse distance weighting, naturally generalizes the basic emulation framework where a single Gaussian distribution is used, to a framework where a mixture of Gaussians is employed. Our results indicate that it might be advantageous to choose a different number of Gaussians in the mixture depending on the goal of the subsequent analysis. For example, is a certain estimation needed or is a confidence interval required? Some interesting future research questions are discussed. Our work is truly interdisciplinary, covering the fields of statistical emulation, demography, agent-based models, genetic algorithms and cluster analysis.

Keywords—Statistical emulation; Agent-based models; Cluster analysis; Genetic algorithms.

I. INTRODUCTION

Statistical emulation, more extensively discussed in Section III-A, is a fairly recent methodology to approximate deterministic computer models. The use of emulation has several advantages, such as the fact that the emulator (the approximation to the computer model) generates its output for a given input almost instantaneously [1], an especially desired property if the considered computer model is computationally expensive. Furthermore, emulation offers a framework that is well suited to perform a sensitivity or uncertainty analysis of the computer model under consideration, see e.g. [2] and [3].

However, the standard emulation framework is not suitable for very large data sets or when stationarity of the data cannot be assumed, as discussed in [4]. The authors of the cited work applied a treed partition model, meaning that they divide up the input space by making binary splits on the value of a single variable so that partition boundaries are parallel to coordinate axes. Partitioning is hierarchical, in the sense that each new partition is a sub-partition of a previous one.

We elaborate on the idea of constructing different emulators for different parts of the input space, but instead of using hierarchical partitioning we apply a non-hierarchical clustering algorithm, namely k-means (see Section III-B). The work that is discussed here is mainly of an exploratory nature, where we experiment with six different implementations. First, we compare results in terms of two alternative definitions of

the distance of a non training data point to a given cluster, where we consider the possibilities of defining this distance as the minimum over all distances from the given point to the training data points belonging to this cluster versus adopting the commonly used distance concept in cluster analysis where the distance from a point to a cluster is taken as the distance to the center of this cluster. Secondly, results are compared in terms of the number of clusters, or equivalently the number of emulators, that are used in obtaining an output for a non training data point. That is, we consider the possibility of assigning a non training data point to more than one cluster, quite similar to fuzzy clustering [5].

The paper is organized as follows. In Section II we describe our agent-based model for which we have constructed an emulator. Section III discusses the three methodologies that play a role in this construction: statistical emulation, cluster analysis and genetic algorithms. Several choices on implementation details have to be made. This results in six different implementation methods, described in Section IV. Section V gives a description of the measures that are used to evaluate the results of these implementation methods. Results are then analyzed in Section VI.

II. APPLICATION TO DATA GENERATED BY AN AGENT-BASED MODEL

A. Short introduction to agent-based models

An agent-based model (ABM) is a computer model that simulates the behavior and interactions of autonomous agents [6]. A key feature is that *population* level phenomena are studied by modeling the interactions of *individuals* that are members of the population of interest. The macro level behavior that emerges from such individual, i.e. lower level, interactions is typically complex, more often than not to the extent that the behavior on the macro scale cannot be deduced by knowledge of the system's parts alone. Thus, for example, an ABM is very well suited to study traffic jams, since a traffic jam results from the behavior of and interactions between individual vehicle drivers, but it may be moving in the direction opposite that of the cars that cause it [7].

B. Description of our agent-based model

We developed an agent-based model to analyze the effectiveness of family policies under different assumptions regarding the social structure of a society. The agents represent the female partner in a household. They are heterogeneous with respect to age, household budget, parity, and intended fertility. A network of mutual links connects the agents to a small subset of the population to exchange fertility preferences. The agents are endowed with a certain amount of time and money which they allocate to satisfy their own and their children's needs. We consider two components of family policies: 1. the policy maker provides a fixed amount of money or monetary equivalent per child to each household and 2. a monetary or nonmonetary benefit proportional to the household income is received by the household. The output on the aggregate level that is produced by the ABM consists of the cohort fertility, the intended fertility and the fertility gap. The inputs that are provided to the ABM include the level of fixed and income dependent family allowances, represented by the parameters b^f and b^v , and parameters that determine the social structure of a society, such as a measure for the agents' level of homophily α , and the strength of positive and negative social influence, which are denoted as pr_3 and pr_4 resp. Analysis of the model outcomes indicates that both fixed and income dependent child supports have a positive and significant impact on fertility, but that the effectiveness of such family policies strongly depends on the social structure of the considered country. This has been described extensively in previous work [8].

C. Data set

The input variables of our ABM are given equidistant values from the input domain and the ABM is applied to generate the corresponding outputs. A total of 741,312 data points in input space were considered in [8]. However, since it is not our purpose here to use emulation to enhance the analysis of the ABM outcomes, but rather to explore the application and behavior of emulation when the training data set has been partitioned into clusters, we limit the size of the data set as follows. First, on the output side we consider only cohort fertility. Secondly, on the input side we consider only those variables that were found to have the largest influence on the outcomes, namely b^f , b^v , α , pr_3 and p_4 . This reduces the data set to 11,232 data points. From this data set we selected randomly 500 points to be used as test points, leaving 10,732 data points as training data set. This is still a very large data set which suffices for our purposes. In this way, we limit computation time to some extent without affecting our objectives.

III. METHODOLOGY

A. Statistical emulation

Statistical emulation provides an approximation to a deterministic computer model that maps real vectors to real vectors. Such a computer model can be seen as a function ν that generates output $y = \nu(\mathbf{x})$ for any given vector \mathbf{x} that belongs to the input domain. In our case study, y is a real number.

The approximation to ν , i.e. the emulator, is determined via the Bayesian formalism as follows. In the first step, it is assumed that nothing is known about ν . The value $\nu(\mathbf{x})$ for any \mathbf{x} is then modeled as a Gaussian distribution with mean $m(\mathbf{x}) = \sum_{i=1}^q \beta_i h_i(\mathbf{x})$, where β_i are unknown coefficients and where h_i represent regression functions chosen by the

user. As is custom in statistical emulation, we choose h_i linear. The covariance between $\nu(\mathbf{x})$ and $\nu(\mathbf{x}')$ is modeled as $\text{Cov}(\nu(\mathbf{x}), \nu(\mathbf{x}') | \sigma^2) = \sigma^2 c(\mathbf{x}, \mathbf{x}')$, where σ^2 denotes a constant variance parameter and where $c(\mathbf{x}, \mathbf{x}')$ denotes a function that models the correlation between $\nu(\mathbf{x})$ and $\nu(\mathbf{x}')$. We adopt the most common choice for c :

$$c(\mathbf{x}, \mathbf{x}') = \exp \left[- \sum_i \left((x_i - x'_i) / \delta_i \right)^2 \right]$$

with x_i and x'_i the i th component of \mathbf{x} and \mathbf{x}' resp., and where the δ_i represent parameters, commonly called the correlation lengths. One way to optimize δ_i is via maximum likelihood [9]. In the second step, training data $(\mathbf{x}_1, \nu(\mathbf{x}_1)), \dots, (\mathbf{x}_n, \nu(\mathbf{x}_n))$ are obtained and, via appropriate applications of Bayes' rule, used to determine the parameters involved and to update the Gaussian distributions to Student's t-distributions. However, it is custom to approximate each t-distribution with a Gaussian distribution. The mean $m^*(\mathbf{x})$ and variance $v^*(\mathbf{x})$ of this distribution, given an input point \mathbf{x} , are produced by the trained emulator. The value $m^*(\mathbf{x})$ is then considered as the approximation for $\nu(\mathbf{x})$ and we determine a 95% confidence interval $CI(\mathbf{x})$ as $[m^*(\mathbf{x}) - 2\sqrt{v^*(\mathbf{x})}, m^*(\mathbf{x}) + 2\sqrt{v^*(\mathbf{x})}]$. We refer to [10] for a more detailed account on emulation. We adopt the widespread practice of considering the distributions that are produced by the trained emulator as Gaussian. This is legitimate, given the sufficiently large number of training data points we use for training.

The correlations between each pair of training data points are encapsulated in a correlation matrix A . The inverse of A is used in the analytical expression for the optimal values of the β_i and of σ^2 . However, when the size of the training data is large it is typically found that A is computationally non-invertible. A simple and widely applied solution is then to replace A by the matrix $A + \alpha I$, where I is the identity matrix and where $\alpha > 0$ is a parameter called the nugget [11].

B. Cluster analysis

Cluster analysis is the unsupervised partitioning of a data set into groups, also called clusters, such that data elements that are member of the same group have a higher similarity than data elements that are member of different groups. Similarity is typically expressed in terms of a user-defined distance measure, such as the commonly used Euclidean distance which we employ here. The best known clustering algorithm is k-means, an iterative algorithm to compute the centers of a predefined number of clusters such that each given data point is assigned to the cluster with the nearest center [5]. We use k-means to subdivide the very large training data set into clusters and then construct an emulator for each of these clusters. Indeed, it is practically impossible to apply emulation directly to the full training data set, as we found that it is even impossible to store in memory the correlation matrix, which contains a number of elements that is of quadratic order in the training data size. Even on much more powerful computers, it is not advisable to train one emulator for the complete training data set, due to computational instabilities in inverting the correlation matrix (see Section III-A). Furthermore, constructing different emulators for different subdomains is a more flexible approach compared to the standard emulation framework, since it allows the parameters β_i and σ^2 to vary from subdomain to subdomain.

C. Genetic algorithms

Determination of the parameters β_i and σ^2 is simple, as analytical expressions exist for their optimal values (see, e.g., [10]). However, such expressions do not exist for the correlation lengths. These are typically obtained by applying the maximum likelihood principle, as described in [9]. The optimization of their density function is a nontrivial task here as this function is a $\mathbb{R}^5 \rightarrow \mathbb{R}$ mapping (there are 5 correlation lengths, one for each of the input variables b^f, b^v, α, pr_3 and pr_4), which may have many local optima. We use genetic algorithms for this optimization task.

Genetic algorithms are a type of heuristic optimization method that mimics some aspects of the process of natural selection, in that a population of candidate solutions to an optimization problem is evolved toward better solutions. Many possible implementations have been developed. The variant we use is essentially the same as described in [12], Section 2.3, and may be summarized as follows:

- 1) A fitness function is chosen by the user. This fitness function measures the quality of a candidate solution. We simply choose the fitness of a vector of correlation lengths as the value of the density function in this vector. The larger the fitness value, the higher the quality of the candidate solution.
- 2) An initial population of candidate solutions is chosen, randomly selected from the input space. The population size is chosen as 100.
- 3) Select the worst half of the population, i.e. the 50 candidate solutions with the lowest fitness value. This population is transformed in the following way. Select repeatedly pairs of different candidate solutions from this subpopulation. Each such pair, say \mathbf{v} and \mathbf{w} , representing vectors of correlation lengths, is changed into a new pair as follows. Generate a random, uniformly distributed $0 < \beta < 1$ and a random number $l \in \{1, \dots, 5\}$ where each of numbers 1, ..., 5 have equal probability to be selected. Then change the value of the l th component v_l and w_l of \mathbf{v} and \mathbf{w} resp. into new values v'_l and w'_l via the following rule:

$$\begin{aligned} v'_l &= (1 - \beta)v_l + \beta w_l \\ w'_l &= (1 - \beta)w_l + \beta v_l \end{aligned}$$

This process is called crossover and amounts to exploring the search space. The crossover process is applied to 25 such pairs.

- 4) Combine the transformed worst half of the population with the unchanged best half into a new population.
- 5) Each of the candidate solutions in the new population possibly undergoes a mutation. That is, each candidate solution is considered in turn and with a certain fixed probability, called the mutation rate, its value (i.e. each of the five components) is changed to a random new value. The mutation rate is typically chosen small; we choose it as 0.05. The purpose of mutation is to maintain diversity in the population and thus to avoid that the algorithm converges to a purely local optimum.
- 6) Go back to the third step unless a predefined maximum number of iterations has been reached. We choose this maximum number as 100. If the maxi-

imum number of iterations is reached, the candidate solution that has the largest fitness value of all candidate solutions, over all 100 iterations, is selected as the optimal vector of correlation lengths.

Key advantages of genetic algorithms are that they only employ function evaluations (and thus not, e.g., information about derivatives as is required by many other optimization methods, such as, for example, gradient descent) and that they are well suited to avoid getting stuck in local optima. Both characteristics make them particularly useful to optimize the density function of the correlation lengths.

IV. DESCRIPTION OF THE SIX IMPLEMENTATION METHODS

Given a set of emulators, each trained on a different part of the input domain, it has to be decided how this set is used to determine an approximate value $\hat{\nu}(\mathbf{x})$ to $\nu(\mathbf{x})$ given a non training data input point \mathbf{x} . First, we have to choose whether we use just one emulator or we combine the outputs of several emulators into one suitable approximate value $\hat{\nu}(\mathbf{x})$. Secondly, in order to choose which specific emulator or emulators to employ we have to make a choice on the definition of the distance from \mathbf{x} to any given cluster, since it is intuitive to use the emulator(s) that correspond to the cluster(s) to which \mathbf{x} is closest.

A. One emulator versus several emulators

Given a single emulator that is to be used to determine an approximate output value in \mathbf{x} , we simply take the posterior mean $m^*(\mathbf{x})$ of this emulator as $\hat{\nu}(\mathbf{x})$ and we use the corresponding confidence interval $CI(\mathbf{x})$ produced by this emulator to model the uncertainty in using $\hat{\nu}(\mathbf{x}) = m^*(\mathbf{x})$ as the true output.

Alternatively, we could use more than one emulator to determine $\hat{\nu}(\mathbf{x})$. This idea is inspired by the use of ensemble methods in machine learning, a technique by which a new data point is classified by taking a weighted vote of the predictions of several classifiers. Such ensembles often perform better than any single classifier [13].

Given an input point \mathbf{x} , the question is then how to determine $\hat{\nu}(\mathbf{x})$ on the one hand and how to obtain an appropriate confidence interval on the other hand. To be more concrete, consider the k emulators that correspond to the k closest clusters, where distance is measured according to either the minimum or the average distance, as explained in the next section. The distances to these clusters are denoted as d_1, \dots, d_k , and the posterior means of the corresponding emulators as $m_1^*(\mathbf{x}), \dots, m_k^*(\mathbf{x})$. We apply an intuitive method to obtain $\hat{\nu}(\mathbf{x})$, called inverse distance weighting, which approximates an unknown value in a given input point by a weighted average of known values in training data input points where the weights are inversely related to the distance from the considered input point to each of the training data points [14]. Using this technique, we determine $\hat{\nu}(\mathbf{x})$ as

$$\hat{\nu}(\mathbf{x}) = \frac{\sum_{i=1}^m \frac{1}{d_i} m_i^*(\mathbf{x})}{\sum_{j=1}^m \frac{1}{d_j}} \quad (1)$$

To determine an appropriate confidence interval, we notice that (1) in fact expresses the mean of a mixture of Gaussian distributions. Thus it is intuitive to obtain the confidence interval from the cumulative distribution function of this mixture. This can be done using the regula falsi method [15].

We have thus shown that combining an idea from machine learning, namely ensemble methods, with an idea from the domain of interpolation, namely inverse distance weighting, leads to a natural extension of the basic emulation framework where instead of using a single Gaussian distribution to model an output a mixture of Gaussian distributions is employed.

B. Center distance versus minimum distance

The k emulators that are used to determine $\hat{\nu}(\mathbf{x})$ and to which we referred in the previous section, are chosen as the emulators that are trained with the k clusters of the training data set that are closest to \mathbf{x} . This requires to define the distance from an input point \mathbf{x} to a cluster C , denoted as $d(\mathbf{x}, C)$. We consider two possibilities, to which we refer as the center distance and the minimum distance resp.

First, $d(\mathbf{x}, C)$ can be defined as the distance from \mathbf{x} to the center of C . Intuitively speaking, the center of a cluster is the data point (not necessarily belonging to the training data set) that is most representative of this cluster. The centers are given as part of the output of the k-means algorithm. This definition of distance from a point to a cluster is commonly used in cluster analysis.

Secondly, $d(\mathbf{x}, C)$ can be defined as the minimum distance from \mathbf{x} to each of the training data points belonging to C . Such a definition conforms the definition of distance from a point to a set in a metric space.

Both definitions rely on the distance between two points, which we take here as the commonly used Euclidean distance.

C. The resulting implementation methods

We obtained results, described below, for $k = 1, 2, 3$. Combining this with the two possibilities for the definition of distance, this gives six alternative methods to determine $\hat{\nu}(\mathbf{x})$ in each of the test points \mathbf{x} . We refer to the method using $k \in \{1, 2, 3\}$ emulators and using the minimum distance as E_{k_min} and to the method using $k \in \{1, 2, 3\}$ emulators and using the center distance as E_{k_cen} . For example, the method where $\hat{\nu}(\mathbf{x})$ is determined using the two emulators that correspond to the two closest clusters, where distance is measured according to the center distance, is denoted as E_{2_cen} .

V. DESCRIPTION OF THE EVALUATION MEASURES

Each of the implementation methods described in the previous section determine a value $\hat{\nu}(\mathbf{x})$ as approximation for $\nu(\mathbf{x})$ and a confidence interval $[l(\mathbf{x}), u(\mathbf{x})]$ around $\hat{\nu}(\mathbf{x})$. The 500 test points are used to evaluate each of the implementation methods. We use two measures to perform this evaluation, one to evaluate the produced approximation to $\nu(\mathbf{x})$, called the average relative difference, and one to evaluate the produced confidence interval, called the average interval score.

A. Average relative difference

Given a test input point \mathbf{x} with corresponding true output $\nu(\mathbf{x})$ and an approximation $\hat{\nu}(\mathbf{x})$ produced by one of the six methods, we can evaluate the quality of the approximation as the relative difference between $\nu(\mathbf{x})$ and $\hat{\nu}(\mathbf{x})$ as follows:

$$RD(\mathbf{x}) = \left| \frac{\hat{\nu}(\mathbf{x}) - \nu(\mathbf{x})}{1/2(\hat{\nu}(\mathbf{x}) + \nu(\mathbf{x}))} \right|$$

The average relative difference for a particular method, denoted ARD , is then the average of $RD(\mathbf{x})$ over all test points \mathbf{x} .

B. Average interval score

The quality of a confidence interval $[l(\mathbf{x}), u(\mathbf{x})]$ around $\hat{\nu}(\mathbf{x})$ is evaluated using the interval score described in [16]. Given an $(1 - \alpha)\%$ confidence interval $[l(\mathbf{x}), u(\mathbf{x})]$, with $\alpha = 0.05$ chosen in this paper, the interval score is defined as

$$IS(\mathbf{x}) = \left(u(\mathbf{x}) - l(\mathbf{x}) \right) + \frac{2}{\alpha} \left(l(\mathbf{x}) - \nu(\mathbf{x}) \right) \mathbf{1}_{\{\nu(\mathbf{x}) < l(\mathbf{x})\}} + \frac{2}{\alpha} \left(\nu(\mathbf{x}) - u(\mathbf{x}) \right) \mathbf{1}_{\{\nu(\mathbf{x}) > u(\mathbf{x})\}}$$

where $\mathbf{1}_{\{expr\}}$ refers to the indicator function, being 1 if expression *expr* holds and 0 otherwise. This scoring rule rewards narrow intervals, while penalizing lack of coverage. The lower its value, the higher the quality of the confidence interval is considered. The average interval score, denoted AIS , is simply the average of $IS(\mathbf{x})$ over all test points \mathbf{x} .

VI. RESULTS

A. Clustering the training data set

In clustering the training data set containing 10,732 data points we have to find a compromise between the desire to obtain clusters with a large enough number of data points to ensure that enough information is available to determine suitable parameter values for the emulator, and the desire to have clusters with a small enough number of data points to avoid computational instabilities (see also Section III-A). Although a nugget can help to avoid computational instabilities (see again Section III-A), it should be used with caution, since the nugget introduces additional uncertainty in an ad hoc way which might not be justified. In any case, the nugget should be smaller than 1, so that the introduced uncertainty is smaller than the process uncertainty [17]. As a rule of thumb, we do not want the nugget to be larger than 0.25 for our emulators. Intuitively speaking we expect that the larger a cluster the larger the nugget has to be to avoid computational instabilities. We determine an appropriate maximum cluster size by randomly selecting subsets of the training data set of varying sizes, and playing with several values for the nugget. This trial and error exercise indicated that for a size of 500 data points a nugget of 0.25 is sufficient to avoid any computational instability. Significantly larger sizes require a larger nugget than 0.25, while a size of 500 with significantly smaller nuggets results in computational instabilities. To illustrate this last point, by choosing the nugget 0.2 we found that there are subsets of the training data set of size 500 for which our software written in R outputs an infinite likelihood, for some values of the δ_i parameters.

A maximum cluster size cannot be given as input to the k-means algorithm, thus we find appropriate clusters by trial and error. Different numbers of clusters are given as input to the k-means algorithm and the resulting number of data points in each cluster is inspected. We find that choosing 32 clusters comes close to the requirement of a maximum cluster size of 500: all but 2 clusters contain at most 500 data points. The 2 largest clusters, having sizes 648 and 652, are split into 2 smaller clusters by applying k-means to each of these clusters and requiring 2 clusters as output for each of them. We then end up with 34 clusters where the largest cluster has size 500.

B. Obtaining an emulator for each cluster

For each cluster an emulator is trained. Although each emulator is trained on a much smaller data set than the full

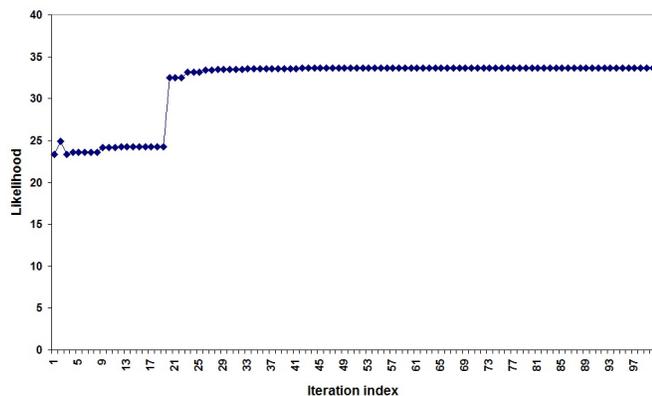


Figure 1. Application of genetic algorithm to find the correlation lengths (first illustration)

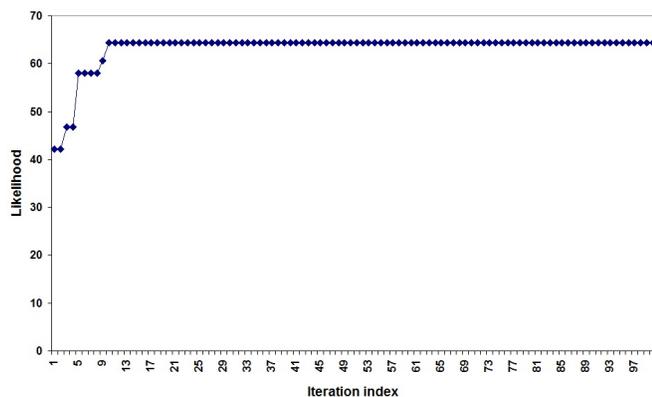


Figure 2. Application of genetic algorithm to find the correlation lengths (second illustration)

training data set, a nugget is still required. The nugget for each of the emulators is chosen heuristically, taking into account the observation that we found subsets of the training data set of size 500 for which a nugget of 0.25 is sufficient to avoid computational instabilities while a significantly smaller nugget does not work (see previous section). If the cluster size is smaller than 150, the nugget is chosen as 0.1, for a cluster size between 150 and 300 the nugget is taken as 0.15, for a cluster size between 300 and 400 as 0.2, and for cluster sizes larger than 400 as 0.25. Computational instabilities are not encountered for any emulator.

As described in Section III-C, a genetic algorithm is used to determine optimal values for the correlation lengths. Figures 1 and 2 show the evolution of the largest fitness value in the population over the 100 iterations for two of the emulators. The fast convergence is also observed for the other emulators.

C. Evaluation of the six implementation methods

The six implementation methods, described in Section IV, are evaluated as explained in Section V. The results are shown in Table I.

Considering ARD, i.e. the evaluation of how well $\hat{\nu}(\mathbf{x})$ approximates $\nu(\mathbf{x})$, we see that E_{1_min} is the best method, with a second place for E_{2_min} . It is also seen that using the minimum distance gives better results than using the center

TABLE I. EVALUATION OF THE SIX IMPLEMENTATION METHODS

Implementation method	ARD	AIS
E_{1_min}	0.14	3.27
E_{1_cen}	0.17	3.95
E_{2_min}	0.21	2.06
E_{2_cen}	0.23	3.21
E_{3_min}	0.24	2.53
E_{3_cen}	0.26	3.52

distance.

Analysis of AIS, i.e. the evaluation of the quality of the produced confidence intervals, shows another picture. For this measure E_{2_min} is by far the best method, while the second best is E_{3_min} . Again, better results are obtained with the minimum distance than with the center distance.

Thus, our experiment points to the following observations:

- The minimum distance gives for both evaluation measures better results. This is counterintuitive to the widespread use of the center distance in cluster analysis.
- For ARD it holds that the *less* emulators are used to determine an approximate value $\hat{\nu}(\mathbf{x})$ the *better*.
- For AIS it holds that using *several* emulators to determine a confidence interval around $\hat{\nu}(\mathbf{x})$ is *better* than using one emulator.

VII. CONCLUSION AND FURTHER RESEARCH

In this paper we have shown how to apply statistical emulation to a very large data set, generated by an agent-based model, using cluster analysis.

In terms of methodology, the main contribution of our work is in demonstrating how the concept of ensemble methods in machine learning can be combined with the inverse distance weighting interpolation method to generalize the standard emulation framework where a single Gaussian distribution models the output of a deterministic computer model, to a framework where a mixture of Gaussian distributions is used for this modeling process.

In terms of experimental results, the presented work shows that the choice of definition of distance from a point to a cluster is an important one, and that one cannot take the center distance, widely used in cluster analysis, for granted. Furthermore, the results demonstrate that it can be advantageous to consider a different number of distributions in a mixture of Gaussians depending on whether an approximation to the deterministic computer model output in a given input is to be determined or a confidence interval is to be sought. For our experiment we found that the less emulators are used to determine an approximation to the output the better, i.e. using one emulator is preferred, while determining a confidence interval is best done with several emulators. Of course, this is a very preliminary conclusion and is by no means to be considered as a general rule. Further research is definitely needed to shed more light on this.

Other further research includes:

- In our work, we allowed a non training data point to belong to several clusters, in contrast to the training data points which were assigned to a single cluster using k-means. Instead of using k-means to cluster the training data set, one can think of applying fuzzy clustering. However, this will result in the loss of a basic

property of emulation, namely that in the training data points it holds that $m^*(\mathbf{x}) = \nu(\mathbf{x})$. This property will be lost when using fuzzy clustering, since in this case a training data point will belong to several clusters. On the other hand, fuzzy emulation, as we might call it, has the advantage that it can preserve continuity of $\hat{\nu}(\mathbf{x})$ by allowing the membership degrees of an input point \mathbf{x} in each of the clusters to change gradually as \mathbf{x} itself changes gradually. This does not hold for our k-means emulation framework, since points \mathbf{x} and \mathbf{x}' that are close together can belong to two different clusters and thus having outputs determined by different emulators.

- Comparison with treed partitioning could provide additional insights into our method as well as into treed partitioning itself. Treed partitioning has the disadvantage of producing an approximation $\hat{\nu}$ to ν that is not continuous, while fuzzy emulation does not have this deficiency. Possibly, ideas from both methods can be integrated in a fruitful way.
- Methodological further research includes developing methods to find an appropriate number of clusters for the given training data set in a systematic fashion, and developing methods to determine the coefficients in the mixture of Gaussians according to some optimality criterion. After all, the inverse distance weighting method determines the coefficients in a purely heuristic way, without taking into account any evaluation measure.

REFERENCES

- [1] S. Conti and A. O'Hagan, "Bayesian emulation of complex multi-output and dynamic computer models," *Journal of Statistical Planning and Inference*, vol. 140, 2010, pp. 640–651.
- [2] A. Sarri, S. Guillas, and F. Dias, "Statistical emulation of a tsunami model for sensitivity analysis and uncertainty quantification," *Natural Hazards and Earth System Sciences*, vol. 12, 2012, pp. 2003–2018.
- [3] E. Chang, M. Strong, and R. Clayton, "Bayesian sensitivity analysis of a cardiac cell model using a Gaussian process emulator," *PLoS One*, vol. 10, 2015.
- [4] R. Gramacy and H. Lee, "Bayesian treed Gaussian process models with an application to computer modeling," *Journal of the American Statistical Association*, vol. 103, 2008, pp. 1119–1130.
- [5] A. Jain and R. Dubes, Eds., *Algorithms for clustering data*. Prentice Hall College Div, 1988.
- [6] N. Gilbert, Ed., *Agent-based models: quantitative applications in the social sciences*. SAGE Publications, Inc, 2007.
- [7] A. Bazghandi, "Techniques, advantages and problems of agent based modeling for traffic simulation," *IJCSI International Journal of Computer Science Issues*, vol. 9, 2012.
- [8] T. Fent, B. Aparicio Diaz, and A. Prskawetz, "Family policies in the context of low fertility and social structure," *Demographic Research*, vol. 29, 2013, pp. 963–998.
- [9] I. Andrianakis and P. G. Challenor, "The effect of the nugget on Gaussian process emulators of computer models," *Computational Statistics and Data Analysis*, vol. 56, 2012, pp. 4215–4228.
- [10] J. Oakley and A. O'Hagan, "Bayesian inference for the uncertainty distribution of computer model outputs," *Biometrika*, vol. 89, 2002, pp. 769–784.
- [11] R. Gramacy and H. Lee, "Cases for the nugget in modeling computer experiments," *Statistics and Computing*, vol. 22, 2012, pp. 713–722.
- [12] J. Carr, "An introduction to genetic algorithms," url: <https://karczmazczuk.users.greyc.fr/TEACH/IAD/GenDoc/carrGenet.pdf>, 2014.
- [13] T. Dietterich, "Ensemble methods in machine learning," in *Proceedings of the First International Workshop on Multiple Classifier Systems*. Springer-Verlag, 2000.
- [14] D. Shepard, "A two-dimensional interpolation function for irregularly-spaced data," in *Proceedings of the 1968 ACM National Conference*.
- [15] J. Sloan and S. Sinha, "Bayesian predictive intervals for a mixture of exponential failure-time distributions with censored samples," *Statistics & Probability Letters*, vol. 11, 1991, pp. 537–545.
- [16] T. Gneiting and A. Raftery, "Strictly proper scoring rules, prediction, and estimation," *Journal of the American Statistical Association*, vol. 102, 2007, pp. 359–378.
- [17] P. Ranjan, R. Haynes, and R. Karsten, "Stable approach to Gaussian process interpolation of deterministic computer simulation data," *Technometrics*, vol. 53, 2011, pp. 366–378.

Using Simulation Modeling to Create Transdisciplinary Process Models and to Build Basic PLM Tools to Support the Systems Engineering Lifecycle

Germano de Souza Kienbaum

Laboratory for Applied Mathematics and Computing
National Space Research Institute (INPE)
São José dos Campos, Brasil
e-mail: kienbaum@lac.inpe.br

Álvaro Augusto Neto

Science Computer Division
Aeronautics Institute of Technology
São José dos Campos, Brasil
e-mail: alvaro@ita.br

Renato Fernandez

Federal Institute for Education, Science and
Technology (IFSP)
São Paulo, Brasil
e-mail: renato@ifsp.edu.br

Elaino Kelson Teixeira Silva

BPMExpert Engenharia de Sistemas LTDA
São José Campos, Brasil
e-mail: kelson@jetsoft.com.br

Abstract - This work describes the use of simulation modeling methodology to create transdisciplinary reference process models, which can then be transformed into specialised models, making use of the methodologies originated from traditional autonomous disciplines that deal with complex discrete event process problems, namely: Systems (Concurrent) Engineering, Project Management, Business Process Management and Simulation Modeling. In particular, the transformation of the enterprise's systems (concurrent) engineering process lifecycle (an aggregate of product engineering and organization's management processes) into specialised models and their implementation, making use of the appropriate inherent methodology and tools originated from these traditional autonomous study areas, result in integrated applications that can be used as tools to support basic Product Lifecycle Management. These tools can be used to support the execution of essential procedures in the Small and Medium Enterprise's systems (concurrent) engineering process, as an alternative to complete and/or tailored Product Lifecycle Management and Business Process Management support systems, which are usually costly, complex, difficult to customize and to integrate with other legacy software applications, and therefore not suitable for use by Small and Medium Enterprises.

Keywords - Systems Engineering; Project Management; Business Process Management; Simulation Modeling; Transdisciplinary Process Science and Technology

I. INTRODUCTION

Transdisciplinary Process Science and Technology (T-ProST) is a term used in this work to designate a holistic view, which consists of the integration of various disciplines that deal with complex discrete event process models: Systems (Concurrent) Engineering (SE) [3][9][13], Project Management (PM) [18], Business Process Management (BPM) [10][14] and Simulation Modeling (SIM) [15]. T-ProST strives towards improving the modeling and seeking better and integrated engineering and management solutions for complex discrete event problems.

The general goal of the research endeavors on T-ProST has a wide scope and very ambitious nature: to develop a mature and comprehensive transdisciplinary process modeling methodology (T-ProST Framework) and to design and implement its integrated supporting environment for application to a large category of complex discrete event process problems. The problem domain covered by T-ProST is essentially the same one covered by discrete event process modeling and simulation procedures based on modeling representations, such as Activity Cycle Diagrams (ACD) [15], Role Activity Diagrams (RAD) [11] and Workflow Diagrams (Workflows) [19].

The main difference and original contribution between the use of the T-ProST Framework and that of the traditional use of its component disciplines is the creation of a transdisciplinary process model and its use as a common basis for the development of their individual specialised process models. This is done by conducting a systematic modeling procedure designated as Conceptual or Reference Modeling and making use of a graphical notation denominated Unified Lifecycle Modeling Diagrams (ULMD), an innovative diagramming technique originated from a fusion of the aforementioned graphical representations [7][8][12].

This work presents the T-ProST approach as a totally integrated application of their component individual autonomous disciplines and their supporting tools to the design phase of the SE lifecycle. The case study performed and explained here is original and it targets the application of the T-ProST Framework to the Small and Medium Enterprise's (SME) SE lifecycle and SME's BPM lifecycle. These denominations refer to the complete systems engineering lifecycle and the SME's business process management lifecycle, respectively, which include, but is not limited to, its product engineering and production management processes.

The limitation of the scope to the design phase of the SE lifecycle in order to explain the T-ProST approach was made for demonstration purpose only. In fact, it can be applied to

any category of complex discrete event process that can be dealt with the ULMD notation and it is intended to be scalable and capable of being used in a wider context, covering the whole organization’s business process management, be it a SME or normal enterprise.

The reason why one targets SME rather than a broader spectrum is that this segment could benefit the most from the methodology and tools proposed in the T-ProST approach. These enterprises can make use of the software environment created with the T-ProST Framework as an alternative to complete and/or tailored Product Lifecycle Model (PLM) and BPM support systems, which are costly, complex, difficult to customize and to integrate with other legacy software applications, and therefore not suitable for use by SME to support their SE lifecycle.

The rest of this work is structured according to the following. Section II defines some fundamentals concepts of the methodology. Section III describes the T-ProST approach. Section IV identifies the problem and sets the overall scope of the research. Section V presents the conceptual modeling phase and the resulting reference model. Section VI describes the specialised model building and analysis phases, for each of the four disciplines comprised by T-ProST (SE, PM, BPM and SIM). Section VII performs a global assessment on the integrated results obtained from the application of the individual disciplines and their supporting tools. Some issues considered are T-ProST’s domain of applicability and the benefits and limitations of the methodology. Finally, in Section VIII, the Holistic Revision and Conclusions presents a summary of the experience gained (the “lessons learned”) with the development of the T-ProST approach and the application of the T-ProST Framework in some case studies already conducted with the methodology.

II. FUNDAMENTALS OF THE METHODOLOGY

The most fundamental and distinctive concepts in T-ProST are those of the transdisciplinary reference process model and its specific notation, denominated Unified Lifecycle Modeling Diagrams (ULMD), as created by the first author [7][8][12]. The reference process model in ULMD describes the system’s lifecycle process in its essential structural and dynamical features, with the special purpose of serving as a common basis for the development of the additional specialised unidisciplinary models to be implemented in each of the autonomous component disciplines of T-ProST.

The actual ULMD designation and its notation format is an evolution of former versions of the diagrams named Unified Simulation Modeling Diagrams (USMD) or Unified Conceptual Modeling Diagrams (UCMD). USMD/UCMD on their turn were originally created from a mix of both ACDs [15] and RAD networks [11] and they can be used for modeling complex discrete event process problems from the domain classes of any of the study areas mentioned before [7].

Table I shows the ULMD symbols used in this work and their respective meaning, both in its former USMD/UCMD version and in its actual format. The actual format of the modeling notation resembles very much that of the basic set of Business Process Modeling Notation (BPMN) [16] icons

with a few extensions and one could question the need for “yet another type of process modeling notation or graphical diagrams”. In fact, BPMN is widely used and supported by large communities, becoming the “de facto” notation for creating models to be converted into specific proprietary notations and implemented in a variety of proprietary tools offered by different systems’ manufacturers. The use of the ULMD notation, however, is just a first step (the conceptual modeling phase) of a gradual modeling process and one can state that some semantic characteristics of ULMD are non-existent in any of the traditional graphical or textual process model representations, such as ACD, RAD, Workflows or even BPMN. These characteristics turn out to be essential for the type of logic and knowledge represented in the reference model.

TABLE I. ULMD: THE CONCEPTUAL MODELING NOTATION

Designation		Description
USMD / UCMD	ULMD	
		Starting point of the entity’s lifecycle process
		Finishing point of the entity’s lifecycle process
		Simple process/activity
		Macroprocess
	N/A	Queue of entities/resources waiting to enter an activity
	N/A	Queue of artifacts/messages waiting to enter an activity
		Link showing the control flow path carrying any type and number of entities and resources being transferred
		Link showing the control flow path carrying a specific artifact, message or triggering mechanism that needs to be distinguished
N/A		Links used for process synchronization
N/A		The crossing over the borders of the flow of control path carrying any type or number of entities and resources. The borders separate swimming lanes or frames and the symbols depict respectively the send and receive mechanisms
N/A		The crossing over the borders of the flow of control path carrying a specific artifact or message being transferred, which needs to be distinguished. The borders separate swimming lanes or frames and the symbols depict respectively the send and receive mechanisms
		Repositories of entities or resources
N/A		Connector used to direct the flow of control among processes to express routing at junctions (or exclusive, split and join)
N/A		Pads indicating the splitting or joining of paths associated with any type or number of entities, resources, artifacts or messages
N/A		Crossovers of entities or resources between two processes displayed in separate swimming lanes or frames

The ULMD distinctive features start with its high level of abstraction and minimalism, comprising only a dozen of different symbols for expressing the complete structural and dynamics of the agents’ lifecycles as concurrent processes. The diagram logic expresses all-in-one: the hierarchical process decomposition, the variety of agents and messages and their process control flows, the agents lifecycles or roles and the interactions among them. The diagrams are created in the form of encapsulated modules, which can be displaced in

frames or swimming lanes and pools, describing from a very simple process (task) to a very complex one (macroprocess).

Underlying all these characteristics, some of which could still be stated in regard to BPMN, there is an unequalled feature that differentiates ULMD from all other types of existing process modeling diagrams. They are not about pure logical operations and control flows (flowcharts), they are essentially about real time consuming transformation activities performed by real entities and messages (artifacts) exchanged between them. The links correspond to real entities flowing along their lifecycle (solid lines) or artifacts (dotted lines) exchanged between the activities, both used as inputs to other activities. Each class of entity has its specific process lifecycle, and they interact to perform their time consuming common activities, which get into a halt until all required inputs (entities and/or artifacts) are available and the additional internal initial conditions for their execution are satisfied. One special type of condition is external and it is depicted graphically, named triggering mechanism. The control flows are always associated with some type of entity, resource, artifact or triggering mechanism, and they reactivate the processes in a hold at a particular point in time and location, making it progress through its complete lifecycle.

III. THE T-PROST APPROACH

A T-ProST study is performed making use of a framework (T-ProST Framework), consisting of three elements, which constitute the pillars of the methodology: Knowledge Structure, made by the transdisciplinary hierarchical process model created (conceptual or reference model); Implementation Method, for their evolution along their respective lifecycles; and Supporting Tools, used to implement their integrated applications [7].

The Knowledge Structure consists of the hierarchical decomposition of the System’s Lifecycle (SL) process model in layers and is represented by the unified reference model. Each phase or macroprocess of T-ProST can be further decomposed into their component activities and single tasks, evidencing the gradual transformation of the System’s Lifecycle and, in particular, the simultaneous use of the diverse disciplines to build the multifaceted model on which the complete T-ProST study is carried out.

Table II presents the T-ProST approach described as macroprocesses and their respective component activities. The first column shows the Implementation Method or Unified Lifecycle Modeling procedure used to create the specialised models and evolve them along their lifecycles.

The macroprocesses or phases in Table II are designated:

- Phase Zero (Mission Definition/ Strategic Planning);
- Conceptual Modeling Phase (Analysis, comprising Elicitation/ IDEF0 Model/ Reference Model);
- Model Development Phase (Specialised Model Building/ Specialised Model Implementation);
- Model Execution Phase (Design of Experiments/ Experimentation);
- Finishing Phase (Specialised Assessment)
- Global Assessment (Integrated Joint Assessment);
- Holistic Revision

The Phase Zero consists in the system or problem definition and the identification of the study’s objectives, which is performed in one activity divided into two tasks, denominated Mission Definition and Strategic Planning, for generality purposes.

The second phase is the Conceptual Modeling Phase, to build the hierarchical process model in ULMD format, describing the operational processes, corresponding to the real transformations occurring in the system. This makes it easier for the different kinds of users and modelers to understand the system’s behavior, and later on to translate this model logic into their specific type of notations, in order to accomplish the specialised model building and construction phases.

TABLE II. T-PROST’S METHODOLOGY AND ITS USE IN SE

TRANSDISCIPLINARY PROCESS SCIENCE			
Unified Lifecycle Modelling	Phases	Activities	Tasks
	Zero	A0	Mission Definition and Strategic Planning
	Conceptual Modeling	Analysis (A)	Elicitation, IDEF0 Model, Reference Model
	Model Development	Specialised Model Building (B) Specialised Model Construction/ Implementation (C)	B1, B2,B3, B4 C1, C2, C3, C4
	Model Execution	Design of Experiments (D) Experimentation (E)	D1, D2, D3, D4 E1, E2, E3, E4
	Finishing	Finishing (F)	F1, F2, F3, F4
	Global Assessment	General Assessment (G)	General Assessment
	Holistic Revision	Holistic Revision (H)	Holistic Revision
			

The remaining phases are those shown in Table II with their respective denomination and more detailed definition given by the decomposition of its respective macroprocess into its components, consisting of composed or single activities, the last ones named tasks. The four types of Tasks B, C, D, E and F are those related with the four component disciplines of T-ProST (SE, PM, BPM and SIM). They are treated separately during the phases of Model Development, Execution and Finishing (specialised modeling, execution and evaluation), and their results are integrated in the Global Assessment Phase, to yield a thorough assessment of the multifaceted model created making use of the transdisciplinary reference model.

The Implementation Method or Unified Modeling of the System's Lifecycle Process is created by the aggregation of the different worldviews used by the individual disciplines that are part of T-ProST. These disciplines are jointly applied making use of a reference model as a common basis for the uniformization of the system's description, representative of its essential structural and dynamical features. The implementation method comprehends the transformation of the reference model into the specialised models, their implementation, execution, analysis and assessment, initially as units, and later on as aggregate, yielding a complete multidimensional analysis of the system, based on a holistic view that encompasses all individual disciplines applied.

The models are created gradually in the form of layers, the most internal one depicting the transformations associated with the systems engineering process (SE model), followed by the PM model that encompasses and extends these operational processes with those necessary for project management.

The BPM model is built on the top of the PM model and extends it with additional processes, whenever necessary, as for example with the management processes related with the coordination of a portfolio of projects. This gradual modeling and implementation procedure is performed until the outermost layer is built, consisting of all processes associated with the core product development processes, the project management processes and the complementary business management processes by the organization, encompassing its complete product lifecycle. Any of the intermediary single layers can be treated separately and used for analysis purposes, in the traditional way their studies are performed. The denomination of SIM model, however, is reserved for the outermost layer describing the complete SL, which comprises the BPM model, with its internal PM and SE layers, and any the additional features necessary to conduct a complete simulation study of the System of Interest (SoI), be it the Product Lifecycle (PL) or the Enterprise Lifecycle (EL), according to the objectives of the study.

IV. PROBLEM DEFINITION

To conduct the actual case study on the application of the T-ProST approach in the SE's design phase process in this work, one decided to use a modified version of the System Engineering Body of Knowledge (SEBoK) [3] reference model of the domain of interest, that is, the design phase of the SE Project. Before going into details about the Product Lifecycle Modeling (PLM) model, though, it is useful to establish the broad scope of the BPM model making use of the APQC's Process Classification Framework (PCF) [1] to illustrate the complete organization EL reference model.

Figure 1 illustrates the scopes in which the concepts System Lifecycle (Enterprise Lifecycle or Product Lifecycle) and their respective Reference Models are used in this work. The EL reference model is based on a top-down view of the organization, describing it according to a holistic process view, encompassing all EL phases, from strategic vision through product development, deployment, operation and the client management, until final product disposal. In this

context, PL and PLM are considered a part of the EL and BPM, as indicated by part 2.0 of APQC's PCF [1], which shows the macro Develop and Manage Products and Services, comprising the product's engineering and management processes.

Part 2.0 of APQC's PCF [1] presents no detailed description about how these generic processes should look like, apparently because they are left to be defined by each specialised type of industry.

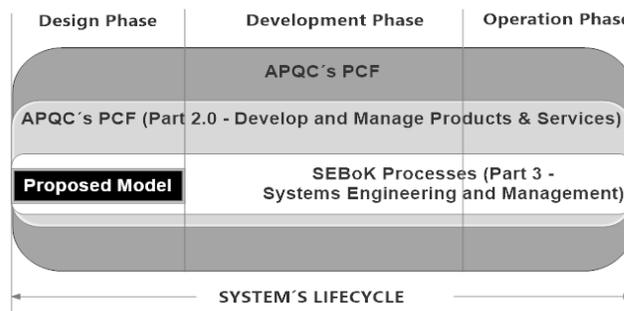


Figure 1. Scopes of APQC's PCF and SEBoK SE's design phase reference models

In the case study presented here this part of the modeling was based on the reference model presented and described in detail in part 3 of the SEBoK [3] and illustrated in Figure 2.

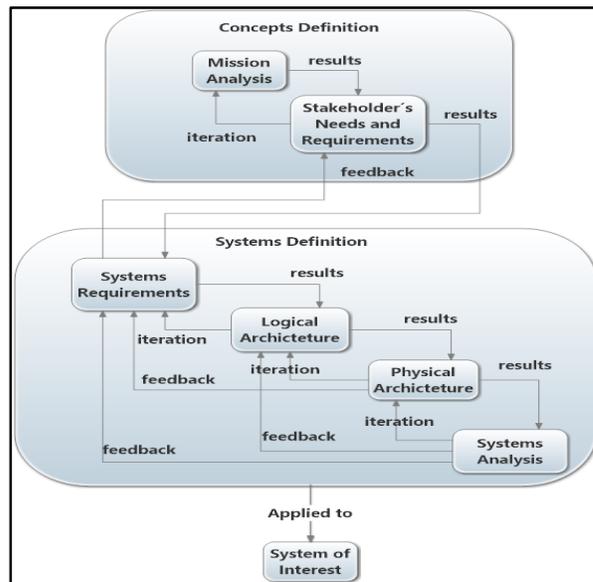


Figure 2. SEBoK [5] SE's design phase process modified by authors Conceptual Modeling

Any enterprise that has SE as the main driver of its production processes may well make use of the PL and PLM concepts definitions in SEBoK [3] to describe their product's design phase according the reference model mentioned above.

The T-PROST approach is demonstrated in the following in a stepwise way, by applying it to the problem described above, according to implementation method shown in column one of Table II.

V. CONCEPTUAL MODELING

The Conceptual Modeling Phase is carried out through an activity denominated Analysis, which consists of three tasks, the first and second ones concerned with the description of the process hierarchical structure (descriptive model), and the third one concerned with its dynamical description (process model).

The descriptive model makes use of techniques typical of the SE worldview: the Elicitation of Information on the process; and the creation of the Integrated DEFINition for function modeling diagrams (IDEF0) [6] to create the IDEF0 model, for process decomposition and documentation. The descriptive model consists in the detailed hierarchical description of the PL in the IDEF0 format. The IDEF0 model comprises all available information on the processes (inputs, outputs, control and resources), except for the workflow of process execution. Figure 3 shows the internal decomposition of the macroprocesses Concepts Definition and Systems Definition in the IDEF0 notation.

The third task of the Conceptual Modeling Phase is typical of the SIM worldview: the Creation of the Reference Model, a process model based on the descriptive model and the addition of the sequence of execution or workflow of

processes. The workflow for process execution is added at this stage of the unified model development lifecycle, making use of the ULMD notation. Despite resorting to mechanisms typical of the SE and SIM worldviews for its elaboration, the reference model created describes the process logic in a very abstract way, which is independent of the specific representations existing in the individual disciplines that are part of T-ProST.

The ULMD reference model of the systems engineering design phase is shown in Figure 4. Two types of loops are used to describe the iterations, which need to be executed in case it is necessary to perform revisions of the work previously accomplished, denominated FB1 and FB2, according to the following definitions:

- FB1 – Iteration: loop for minor revision, between two consecutive processes, belonging to the same macroprocess or phase of a Project.
- FB2 – Feedback: loop between two non consecutive processes, which might belong to a single macroprocess or different ones, for reworking an outcome that should have already been produced in previous stages of the entitie’s lifecycle, but which needs to be redefined or to suffer a major revision.

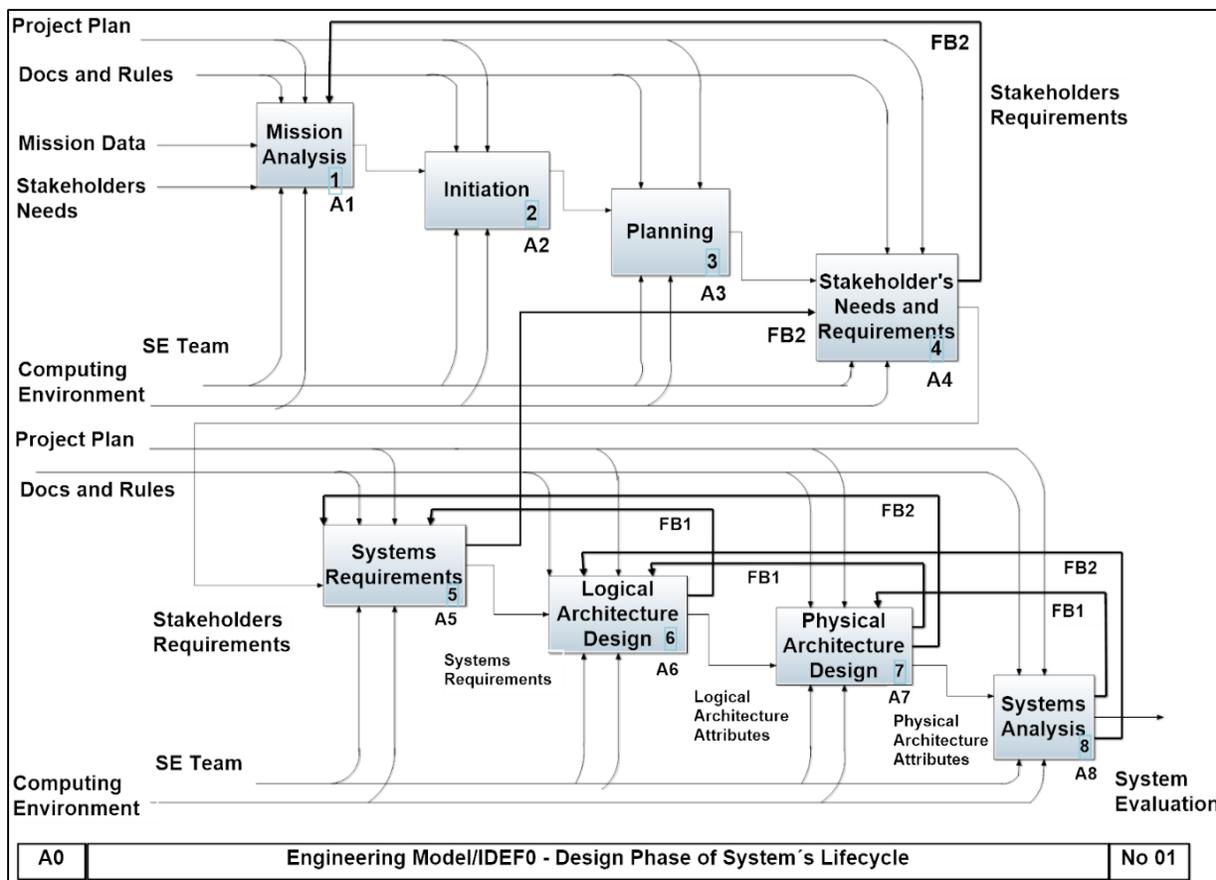


Figure 3. SE’s design phase – concepts definition and systems definition – in IDEF0

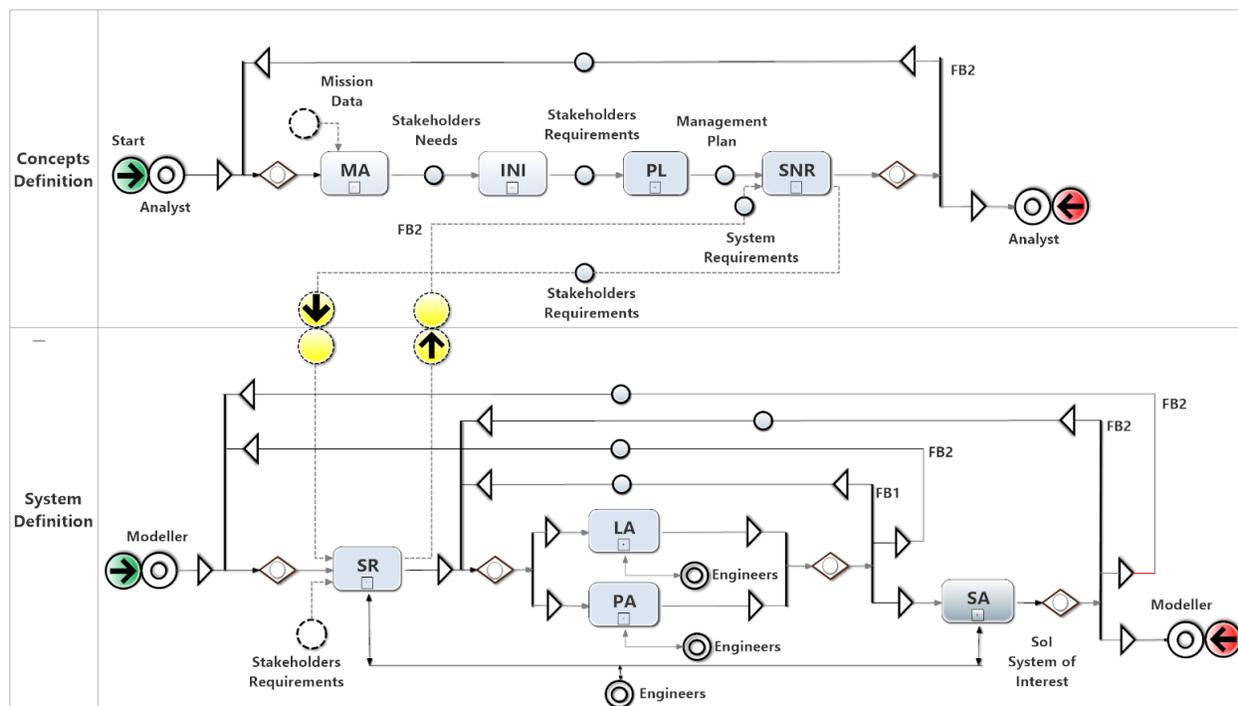


Figure 4. SE design phase – concepts definition and system definition – in ULMD

The macroprocesses Concept Definition and System Definition might undergo successive iterations, during the execution of the systems engineering lifecycle design phase, and the processes might be executed by just one class type of entity or by different types, each class type responsible for a specific process, for example, mission analysts and system modelers. The agents responsible for the execution will act sequentially or in parallel, according to the execution flow previously defined or that established at runtime by means of the scripts used in the activities for routing the control flow during model execution. This is assured by drawing the diagrams as sequential processes only if this is mandatory, otherwise they are displayed as parallel and their execution might occur simultaneously, if there are enough resources available, or they might be executed sequentially, if the conditions and internal limitations make it necessary. A loop encompassing the processes whose sequential or parallel execution is not determined beforehand enforces that all of them are correctly performed before the control flow is passed ahead to the next process in the entity's lifecycle.

VI. SPECIALISED MODEL BUILDING AND ANALYSIS

The Model Development phase is performed according to each of specialised modeling and analysis techniques originated from the autonomous disciplines component of T-ProST.

The transformation of the Reference Model into the specialised models with their supporting applications can be used as a basic PLM environment, if the software tools used possess all the management features required for documenting, monitoring and controlling product's configuration and its evolution along the entire product's

development lifecycle, from conception to disposal. The first steps performed by the SE Modeling tasks B1 and C1 in Table II are just the start for the creation of a more complete and integrated environment. This complete environment will be formed by the addition of the tools created through tasks B2 and C2 in the PM Modeling activity, those created through tasks B3 and C3 in the BPM Modeling activity, as well as those created through the tasks B4 and C4, in the Simulation Modeling activity. All these tools have a complementary nature, resulting from the fact that they all make use of the same reference model and adopt a systematic approach for their implementation, as described below, in the following steps of the application of the T-ProST Framework.

A. SE Modeling

The Systems Engineering Modeling consists in the transformation of the Reference Model into the specialised SE model. This corresponds to thoroughly documenting the model, starting with the known knowledge content about the Reference Model, that is, a merge of the information shown in Figure 3 and Figure 4. The SE modeling might be supported by databases applications and also by some kind of configuration and control management system, ideally one that can display, both graphically and as linked data, the complete information on the model, including the process model hierarchical decomposition and execution sequence, for monitoring purposes.

The documentation of the case study presented in this work required keeping track of all data related to the product design phases (Concepts Definition and Systems Definition), as well as the addition of more model documentation, such as the descriptions of the artifacts created by the SE processes.

B. PM Modeling

The objective of the PM study area is to plan and coordinate the activities necessary to provide a satisfactory product, service or enterprise’s endeavour, within the boundaries set by the chronogram, the budget, resources, infrastructure and human resources and technology available.

The traditional way to describe a project is by representing it as a sequenced network of activities, by means of diagrams

known as Program Evaluation and Review Technique (PERT) [18], a renowned and well documented technique, used for management of engineering projects, be it of a service or industrial nature, aiming at their planning and execution control. This would result at a macroprocess level in the diagram depicted in Figure 5.

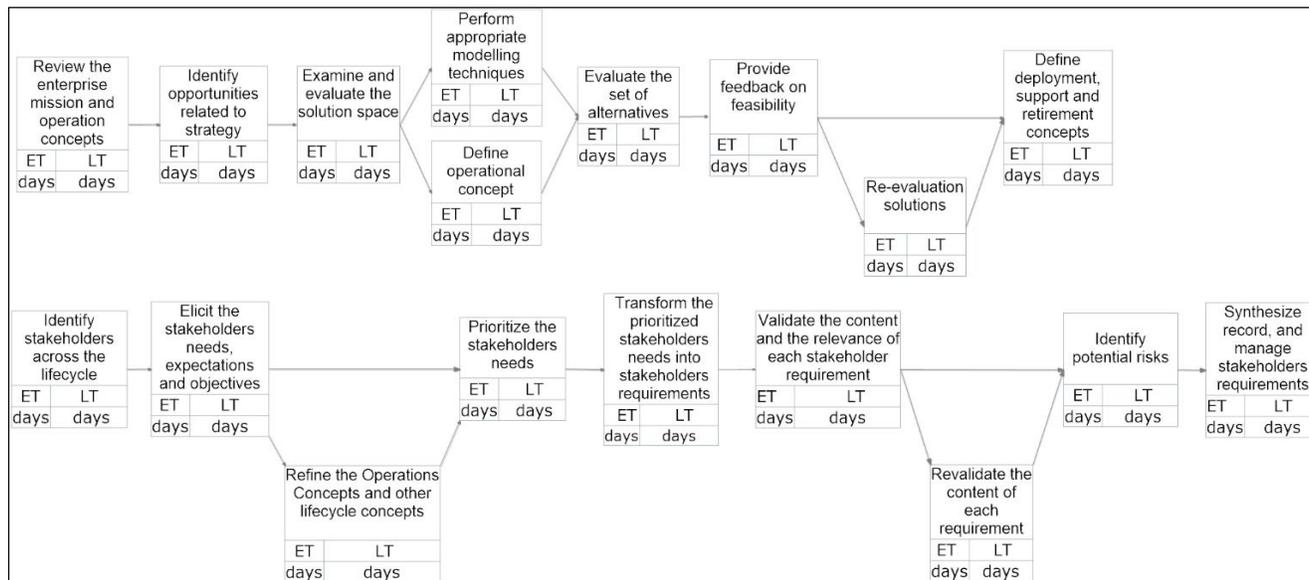


Figure 5. Concepts definition process – PERT model

PM modeling is seen as the application of PM techniques making use of a process oriented worldview supported by software tools to extend the SE model with the necessary activities related with the management of product development along the entire product lifecycle. The PM model extends the SE process model with additional PM processes and explains all relations between these two types of models [7].

By using the T-ProST approach, one can get great gains by addressing PM not just for the management of a single project, but for the management of multi-projects, through process modeling and the integration of the concepts and methodologies used in PM, BPM and SIM. Traditionally, the PM, BPM and SIM methodologies and their supporting tools have remained conceptually very divergent and self-contained in their respective knowledge and application domains, showing little communication between them, due to this “single enterprise” way of handling projects in the PM study area.

On the one hand, project management concepts, such as critical paths and completion time for a segment of the process, typical of PM, can be applied in simulation studies to yield greater productivity and a thorougher analysis of possible strategies for system operation. On the other hand,

simulation modeling concepts, such as idle times of entities staying in queues in front of activities and dynamical resources allocation, can be applied to reduce segments or overall process completion times and costs in PM applications in the SE lifecycle.

C. BPM Modeling

BPM is a structured and systematic approach for conducting modeling, analysis, automatic execution and control, management and continuous improvement of processes used in the product engineering and production management of complex product and services [10][14].

Figure 6 shows the design phase of the systems engineering lifecycle in the BPMN notation [16]. This is just the first step of the modeling process, the model representation using the Bizagi’s Process Modeler [2]. The complete modeling procedure needs a second modeling step, making use of the Bizagi suite, an environment where this initial graphical model can be extended with many additional features and transformed into running applications, which may be used for automating some features of the system’s operational process. The complete set of software tools to support the BPM methodology is known as Business Process Management Systems (BPMS) [10].

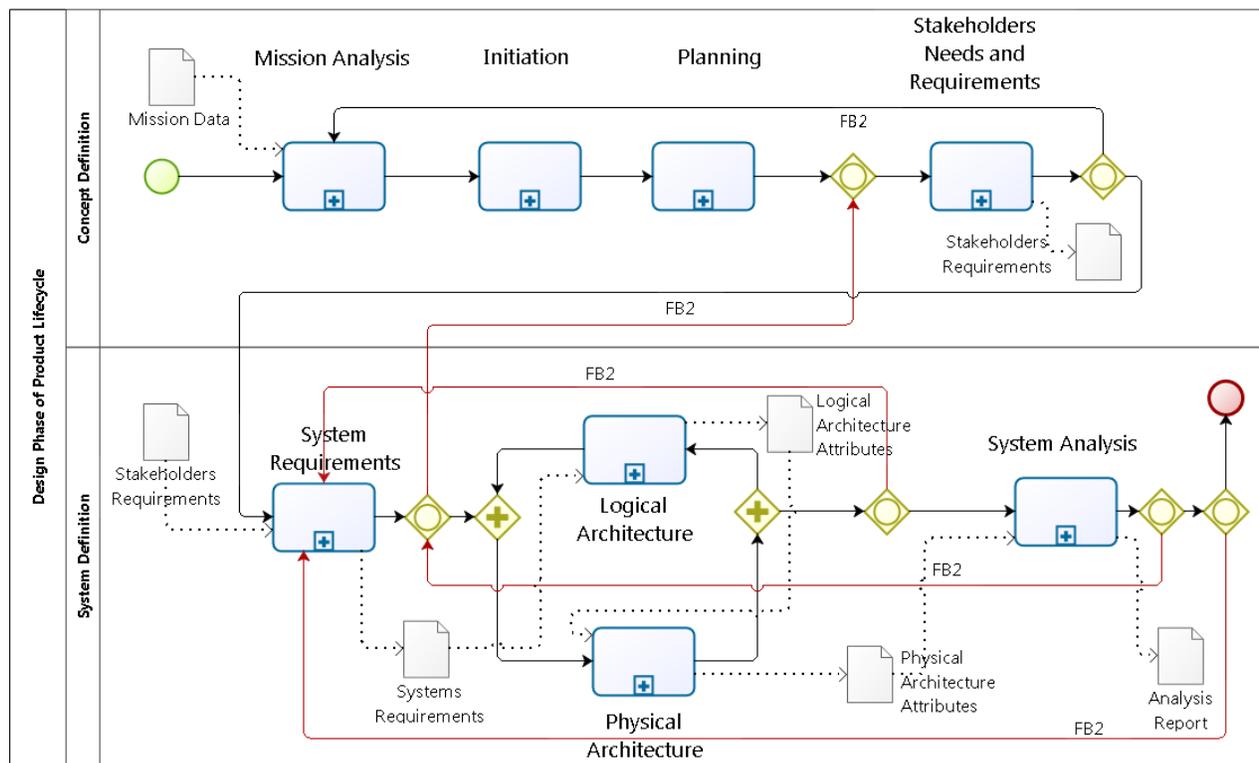


Figure 6. SE's design phase – concepts definition and systems definition – BPMN model

The BPMS tools work throughout the complete model life cycle of business processes studies: modeling, implementation, execution, automation, monitoring, analysis and continuous improvement of the system. They provide an environment for the complete development and deployment of an application that will act as part of the mechanisms used in the operation of the real system, playing a major role in aspects regarding its execution, its monitoring and control, and contributing for its continuous improvement.

The organizations, however, still face a complex scenario for customizing and improving their business processes due to the diversity, incompatibility and/or incompleteness of methods, as well as the high costs of deployment of its supporting tools. The T-ProST general systematic approach could help to integrate and unify concepts and tools used in support of SE, PLM, PM and BPM and to help discard some overhead results that might turn out to be inconsistent, duplicate or superfluous regarding these existing individual study areas. The use of BPMS systems can provide the means for implementation of automated procedures to conduct the operation of the real system. Other main applications to compose the T-ProST supporting environment might be chosen among existing tools as well, and additional interfaces might be built for their integration

D. Simulation Modeling

The use of SIM in T-ProST follows the traditional definition of simulation with the remark that one is addressing the aggregate of the processes representative of the systems

engineering and of the organization management lifecycles [7].

The complete simulation application requires the creation of the facilities needed for the experimentation (design of experiments) and of any other additional features for analysis and presentation of results.

Model assessment in simulation studies can be made by a combination of the normal procedures used in simulation for this purpose with the techniques derived from project management of a multi-project nature, with the aim of enhancing the understanding of the experimental factors and strategies which significantly affect system's productivity.

The integrated use of standard model assessment and project management techniques in simulation studies reveals that they have a complementary nature, the first group of procedures allowing for the analysis of the dynamics of the process, including optimization of resources allocation, and the second one allowing for a better evaluation of activities completion time of partial or complete production cycles, as well as their cost assessment.

The only special feature is the use of the SL reference model, represented in the ULMD notation, to base model development in the proprietary simulation system offered by the different manufacturers. The reference model is transformed into the specific model notation and implemented according to the manufacturers of the proprietary systems, as shown in the example of Figure 7, created with the SIMPROCESS simulation system [5].

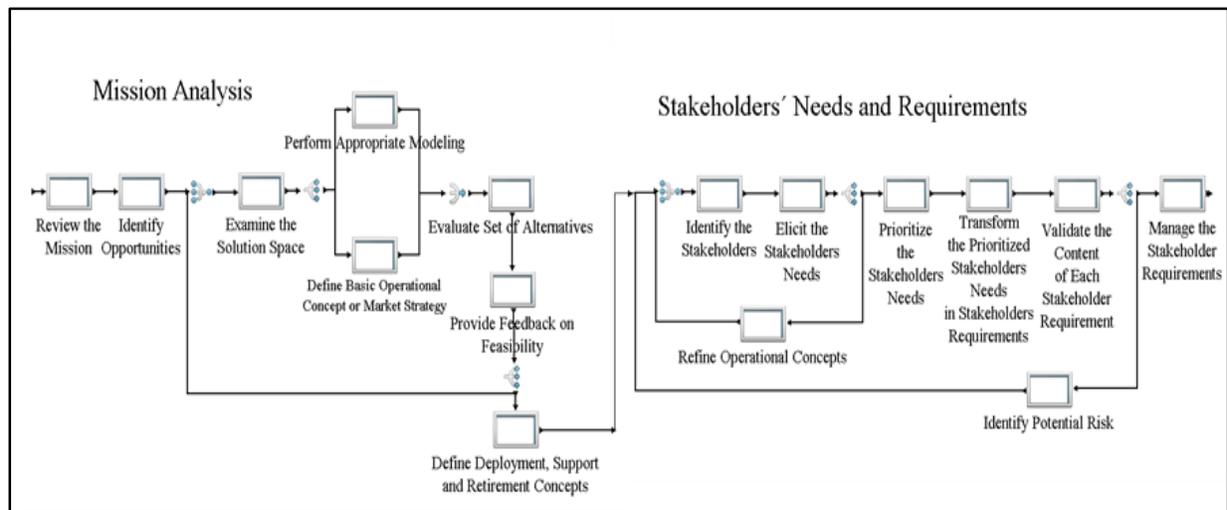


Figure 7. SE'S design phase - concepts definition macroprocess – simprocess model

The application of the existing software tools interchangeably in each of these study areas can already improve the simulation model assessment and the multi-projects management tasks. The combination of these two techniques is therefore very promising, but its advantages cannot be entirely exploited through the separate use of the actual existing software tools of the individual types in studies of the other area, because they have been designed with different purposes, which do not take into account their complementary nature.

VII. GLOBAL ASSESSMENT

T-ProST presents an interesting alternative way to improve the solution of problems involving complex business processes, that is, instead of making use of the existing knowledge base to create new techniques and tools, and/or to improve the already existing ones, one would focus on the (re)structuring, integrating and unifying the knowledge base underlying them.

This holistic approach could consolidate all the scattered knowledge on complex process under a single umbrella, capable of covering the entire problem domain and covering the complete process model life cycle. This innovative holistic view addresses both the restructuring of the knowledge content (descriptive view) and the creation of a general systematic modeling procedure (unified process view), making use of reference models to base the individual or combined use of the diverse modeling techniques to create multifaceted models.

The objective is to progress towards the integration and the unification of the concepts, methods and tools from these diverse techniques, seeking to apply them simultaneously in an easy and consistent way, thereby achieving the same complementary benefits from their individual use and, at the same time, avoiding any overheads, inconsistencies and duplications related with their joint application.

The proposed methodology can be applied to a large problem class, with special interest focused on applications in

serial processes and in multi-projects of an identical single project nature, whose execution share common resources and occur in parallel.

The full development and application of the methodology will require the use of existing tools to perform several case studies, as well as the creation of a new hybrid environment, which on its turn will require quite a lot of software application and development time. For the moment, one has already performed studies making use of the following support tools: ProjectLibre [17], to support project management applications; Bizagi [2] and Bonita [4], to support the application of BPM techniques; and Simprocess [5], to build simulation models and conduct their analysis of results. The main reason for choosing these tools was their availability in the form of open source or low cost academic licenses. They can be easily substituted by similar systems, since the great majority of them can cope with the implementation of the specialised models described in the SE, PM, BPM and SIM categories presented in this work.

There is an expected overhead in this attempt to apply a transdisciplinary approach based on existing autonomous tools, represented by the expensive procedure in regard to maintaining model consistency and compatibility across the whole model development life cycle. This overhead resulting from the application of a unified approach right from the start of the modeling process is seen rather as an anticipation of future problems, which might remain unsolved if the traditional way, based on independent application of the techniques and the gathering and interpretation of their results to build a global solution thereafter, is applied. This drawback can be reduced in future studies of this kind by the use of formal verification procedures and of software mechanisms to improve process model consistency and compatibility.

The authors advocate further that a more encompassing and definitive solution for this problem can be achieved through the evolution of the T-ProST Framework and the improvement of its supporting tools, aiming at its transformation into a mature methodology. The foundations

and guidelines for implementation of this methodology have been presented in this work and they will be pursued and extended in future research.

VIII. HOLISTIC REVIEW AND CONCLUSIONS

The application domain of the T-ProST approach comprises not only the Product's Lifecycle (PL) processes with its development and management processes (known as PLM in the literature), but also the Enterprise's Lifecycle (EL) process, with its business process management procedures (known as BPM in the literature). The complete holistic view on processes advocated by the T-ProST approach considers PLM as a part of BPM, and refers to product's and enterprise's lifecycles simply as System's Lifecycle (SL), with the meaning of the *system* concept made clear by the context in which it appears.

The USMD/UCMD former versions of the notation have been used to describe and conduct some case studies on a variety of discrete event problems, such as a steelworks problem used as a class project, which is an example of a serial production process, typical of the SIM study area. Another example of application was on a house construction process, typical of the PM study area, which has been modeled and studied as a hybrid PM and SIM problem, showing the diversity of systems that can be represented using the notation.

The present work improves the experience obtained in those previous case studies and presents T-ProST as an aggregate of the application of the individual techniques. The multifaceted modeling makes use of a reference model of the design phase processes of the systems engineering lifecycle as a common basis, aiming at the integration and the exploitation of their complimentary advantages. As a result, the T-ProST framework has been revised, which will lead to the development of a mature methodology and its supporting tools for application in additional case studies in the future, for continuous improvement of the methodology and its supporting environment.

The full benefits of a unified methodology shall only be achieved if one undertakes the design and the construction of a hybrid environment to deal simultaneously with all the issues involved in the individual study areas. A hybrid environment based on an integrated approach will allow the modeling and analysis of a production process as the application of basic SE, PM, BPM and SIM tools, in substitution of complex PLM systems, which are costly and difficult to customize, and therefore not adequate for SME. With such tools it will be possible to increase productivity, reducing segment or full process completion time, subjected to resources and costs constraints, by shifting and analysing the start of successive production cycles, for example.

ACKNOWLEDGMENTS

The authors acknowledge the MSc and PhD research students who participated in the courses CSE-326 and CSE-331 year 2015 of the Post-Graduation Program ETE/INPE, who participated in the development and implementation of the system engineering design phase models, as class projects.

REFERENCES

- [1] American Productivity and Quality Center. Process classification framework. [Online]. Available from: <https://www.apqc.org/pcf> 2015.10.15
- [2] Bizagi. BPM suite overview. [Online]. Available from: <http://www.bizagi.com/en/bpm-suite> 2015.10.15
- [3] Bkcase Editorial Board. The guide to the systems engineering body of knowledge (SEBoK), v. 1.4. R. D. Adcock (EIC). Hoboken, NJ: The Trustees of the Stevens Institute of Technology, June 2015. [Online]. Available from: http://sebokwiki.org/wiki/Download_SEBoK_PDF 2015.10.15
- [4] Bonita BPM 7.1.2. Community Edition. [Online]. Available from: <http://www.bonitasoft.com/products-v2#versions-community/2015.10.23>.
- [5] Caci Inc. Federal. Simprocess - Product overview and user's manual. Release 5.2. [Online]. Available from: <http://simprocess.com/about-simprocess/simprocess-documentation/> 2015.10.15
- [6] Federal Information Processing Standards Publications (FIPS PUBS). Integration Definition for Function Modeling (IDEF0). [Online]. Available from: <http://www.idef.com/pdf/idef0.pdf> 2015.10.10
- [7] G. S. Kienbaum, A framework for process science and technology and its application to systems concurrent engineering. São José dos Campos: INPE, Technical Report. [Online]. Available from: <http://urlib.net/8JMKD3MGP3W34P/3KG6NB5> 2015-10-29.
- [8] G. S. Kienbaum, "A framework for process science and technology applied to concurrent engineering", In: ISPE International Conference on Concurrent Engineering, 19., 2012, Trier, Germany. Proceedings... London: Springer-Verlag, 2012. v. 2, p. 1033-1044. ISBN 9781447144250. [Online]. Available from: <http://plutao.sid.inpe.br/col/dpi.inpe.br/plutao/2012/11.28.14.40.58/doc/CE2012.pdf>
- [9] International Council On Systems Engineering. Systems engineering handbook: a guide for system lifecycle processes and activities. 4th Edition. Hoboken, NJ, John Wiley & Sons, 2015.
- [10] J. Jeston and J. Nelis, Business process management: practical guidelines to successful implementations, Oxford: Butterworth-Heinemann, 2008.
- [11] J. Murdoch and J. McDermid, "Modelling engineering design processes with role activity diagrams", In: Transactions of the SDPS, Vol. 4, No. 2, June 2000, pp.45-65.
- [12] L. A. Silva, G.S. Kienbaum, G. Loureiro and M. M. Tanik, "A design and process science study applied to the Laboratory for Integration and Testing of the National Space Research Institute (LIT/INPE)," In: SDPS 2011. 2011, Jeju Island South Korea. Proceedings... [Online]. Available from: <http://plutao.sid.inpe.br/col/dpi.inpe.br/plutao/2011/06.11.14.52/doc/SDPS2011.pdf>.
- [13] NASA. Systems engineering handbook. 2007. [Online]. Available from: <http://ntrs.nasa.gov/> 2015.10.10
- [14] N. Melão and M. Pidd, "A conceptual framework for understanding business processes and business process modelling", In: Information Systems Journal, 2000, pp. 105-129.
- [15] M. Pidd, Computer simulation in management science, 3rd Edition. John Wiley & Sons, Chichester, UK, 1992.
- [16] Object Management Group. Business process modeling notation – V. 2.0. OMG Document Number: formal/2011-01-03. [Online]. Available from: <http://www.omg.org/spec/BPMN/2.0/> 2015.10.10
- [17] ProjectLibre User Guide - Version 0.3.3. [Online]. Available from: <http://www.projectlibre.org/> 2015.10.23.
- [18] Project Management Institute. A guide to the project management body of knowledge, 5th ed. Newtown Square, PA: Project Management Institute, 2013.
- [19] W. Van der Aalst, M. Weske and G. Wirtz, "Advanced Topics in Workflow Management: Issues, Requirements and Solutions", In: Transactions of the SDPS, Vol. 7, No. 1, 2003, pp. 49-77

Simulation Methodologies for the Numerical Analysis of High-Speed Dynamics

Meshing and High-Performance Computing as Influencing Factors

Arash Ramezani and Hendrik Rothe

Chair of Measurement and Information Technology

University of the Federal Armed Forces

Hamburg, Germany

Email: ramezani@hsu-hh.de, rothe@hsu-hh.de

Abstract—By now, computers and software have spread into all fields of industry. Therefore, extensive efforts are currently made in order to improve the safety by applying certain numerical solutions. For many engineering problems involving shock and impact, there is no single ideal numerical method that can reproduce the various aspects of a problem. An approach which combines different techniques in a single numerical analysis can provide the “best” solution in terms of accuracy and efficiency. This paper presents a set of numerical simulations of ballistic tests, which analyze the effects of armor structures. The goal is to find an appropriate technique for simulating composite materials and thereby improve modern armor to meet current challenges. To have the correct material model available is not enough. In this work, the main simulation methodologies are compared to create a perfect model for different armor structures. The calculation should match ballistic trials and be used as the basis for further studies. Given the complexity of penetration processes, it is not surprising that the bulk of work in this area is experimental in nature. Terminal ballistic test techniques, aside from routine proof tests, vary mainly in the degree of instrumentation provided and hence the amount of data retrieved. Here, both the ballistic trials as well as the analytical methods will be discussed.

Keywords—solver methodologies; simulation models; composite materials; high-performance computing; armor systems.

I. INTRODUCTION

In the security sector, failing industrial components are ongoing problems that cause great concern as they can endanger people and equipment. Therefore, extensive efforts are currently made in order to improve the safety of industrial components by applying certain computer-based solutions. To deal with problems involving the release of a large amount of energy over a very short period of time, e.g., explosions and impacts, there are three approaches, which are discussed in detail in [1].

As the problems are highly non-linear and require information regarding material behavior at ultra-high loading rates, which are generally not available, most of the work is experimental and may cause tremendous expenses. Analytical approaches are possible if the geometries involved are relatively simple and if the loading can be described through boundary conditions, initial conditions, or

a combination of the two. Numerical solutions are far more general in scope and remove any difficulties associated with geometry [2].

For structures under shock and impact loading, numerical simulations have proven to be extremely useful. They provide a rapid and less expensive way to evaluate new design ideas. Numerical simulations can supply quantitative and accurate details of stress, strain, and deformation fields that would be very costly or difficult to reproduce experimentally. In these numerical simulations, the partial differential equations governing the basic physics principles of conservation of mass, momentum, and energy are employed. The equations to be solved are time-dependent and nonlinear in nature. These equations, together with constitutive models describing material behavior and a set of initial and boundary conditions, define the complete system for shock and impact simulations.

The governing partial differential equations need to be solved in both time and space domains (see Figure 1). The solution for the time domain can be achieved by an explicit method. In the explicit method, the solution at a given point in time is expressed as a function of the system variables and parameters, with no requirements for stiffness and mass matrices. Thus, the computing time at each time step is low but may require numerous time steps for a complete solution.

The solution for the space domain can be obtained utilizing different spatial discretization techniques, such as Lagrange [3], Euler [4], Arbitrary Lagrange Euler (ALE) [5], or “mesh free” methods [6]. Each of these techniques has its unique capabilities, but also limitations. Usually, there is not a single technique that can cope with all the regimes of a problem [7].

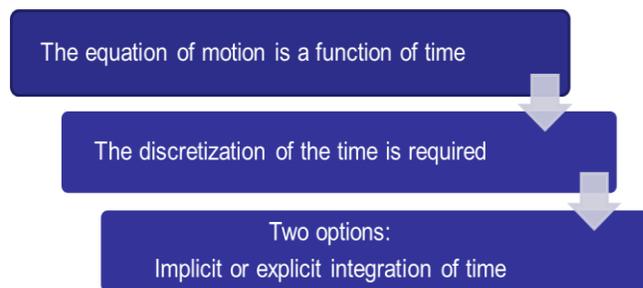


Figure 1. Discretization of time and space is required.

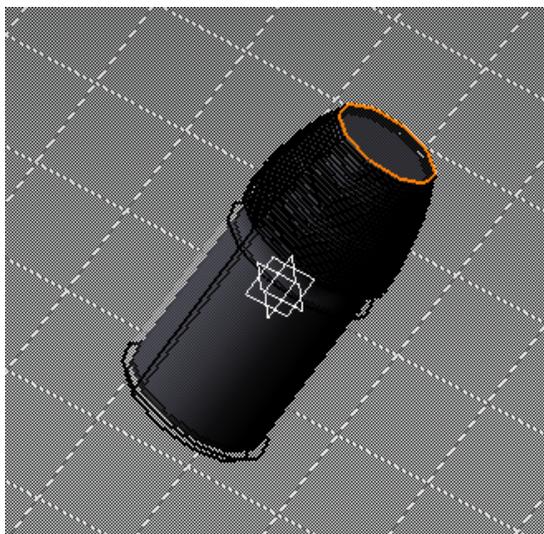


Figure 2. Native CAD geometry of an exemplary projectile.

This work will focus on high-speed dynamics, esp. impact simulations. By using a computer-aided design (CAD) neutral environment that supports direct, bidirectional, and associative interfaces with CAD systems, the geometry can be optimized successively. Native CAD geometry can be used directly without a translation to IGES or other intermediate geometry formats [8]. An example is given in Figure 2.

The work will also provide a brief overview of ballistic tests to offer some basic knowledge of the subject, serving as a basis for the comparison and verification of the simulation results.

The objective of this work is to compare current simulation methodologies to find the most suitable model for high-speed dynamics and impact studies. Lagrange, Euler, ALE, and “mesh free” methods, as well as combinations of these methods, are described and applied to a modern armor structure impacted by a projectile. It aims to clarify the following issues: What is the most suitable simulation model? How does the mesh density affect the results? What are the benefits of high-performance computing?

The results shall be used to improve the safety of ballistic structures, esp. for armored vehicles. Instead of running expensive trials, numerical simulations should be applied to identify vulnerabilities of structures. Contrary to the experimental results, numerical methods allow an easy and comprehensive study of all mechanical parameters.

Modeling will also help to understand how the armor schemes behave during impact and how the failure processes can be controlled to our advantage.

After a brief introduction and description of the different methods of space discretization in Section III, there is a short section on ballistic trials where the experimental set-up is depicted, followed by Section V describing the analysis with numerical simulations. In Section VI, the influence of high-performance computing is emphasized. The paper ends with a concluding paragraph.

II. STATE-OF-THE-ART

Simulating penetration and perforation events requires a numerical technique that allows one body (penetrator) to pass through another (target). Traditionally, these simulations have been performed using either an Eulerian approach, i.e., a non-deformable (fixed) mesh with material advecting among the cells, or using a Lagrangian approach, i.e., a deformable mesh with large deformations. The main point of criticism of the Eulerian approach has been that the shape of the penetrating body, usually an idealized rigid projectile, becomes “fuzzy” as the penetration simulation proceeds, due to the mixing of advected materials in the fixed Eulerian cells. Lagrangian methods require some form of augmentation to minimize or eliminate large mesh distortions. The so-called “pilot hole” technique and the material erosion are the two most often used augmentations for Lagrangian penetration simulations. In the pilot hole technique, elements are removed a priori from the target mesh along the penetrator trajectory, which works well for normal impacts where the trajectory is known a priori. The latter technique removes distorted elements from the simulation based upon a user supplied criterion. They are also removed along the penetrator trajectory, but with no general guidance for selecting certain criteria, i.e., they are ad hoc.

The focus of the present work is to assess a relatively new class of numerical methods, so-called mesh free methods, which offer analysts an alternate analytical technique for simulating this class of ballistic problems without a priori trajectory knowledge or the need to resort to ad hoc criteria. The assessment is made by comparing projectile residual speeds provided by the various techniques, when used to simulate a ballistic impact experiment. The techniques compared are the mesh free method known as Smooth Particle Hydrodynamics (SPH), a multi-material ALE technique, and Lagrangian with material erosion. Given that comparing these inherently different methods is hardly possible, large efforts have been made to minimize the numerous ancillary aspects of the different simulations and focus on the unique capabilities of the techniques.

III. METHODS OF SPACE DISCRETIZATION

The spatial discretization is performed by representing the fields and structures of the problem using computational points in space, usually connected with each other through computational grids. Generally, the following applies: the finer the grid, the more accurate the solution. For problems of dynamic fluid-structure interaction and impact, there typically is no single best numerical method which is applicable to all parts of a problem. Techniques to couple types of numerical solvers in a single simulation can allow the use of the most appropriate solver for each domain of the problem [9].

The most commonly used spatial discretization methods are Lagrange, Euler, ALE (a mixture of Lagrange and Euler), and mesh-free methods, such as Smooth Particles Hydrodynamics (SPH) [10].

A. Lagrange

The Lagrange method of space discretization uses a mesh that moves and distorts with the material it models as a result of forces from neighboring elements (meshes are imbedded in material). There is no grid required for the external space, as the conservation of mass is automatically satisfied and material boundaries are clearly defined. This is the most efficient solution methodology with an accurate pressure history definition.

The Lagrange method is most appropriate for representing solids, such as structures and projectiles. If however, there is too much deformation of any element, it results in a very slowly advancing solution and is usually terminated because the smallest dimension of an element results in a time step that is below the threshold level.

B. Euler

The Euler (multi-material) solver utilizes a fixed mesh, allowing materials to flow (advect) from one element to the next (meshes are fixed in space). Therefore, an external space needs to be modeled. Due to the fixed grid, the Euler method avoids problems of mesh distortion and tangling that are prevalent in Lagrange simulations with large flows. The Euler solver is very well-suited for problems involving extreme material movement, such as fluids and gases. To describe solid behavior, additional calculations are required to transport the solid stress tensor and the history of the material through the grid. Euler is generally more computationally intensive than Lagrange and requires a higher resolution (smaller elements) to accurately capture sharp pressure peaks that often occur with shock waves.

C. ALE

The ALE method of space discretization is a hybrid of the Lagrange and Euler methods. It allows redefining the grid continuously in arbitrary and predefined ways as the calculation proceeds, which effectively provides a continuous rezoning facility. Various predefined grid motions can be specified, such as free (Lagrange), fixed (Euler), equipotential, equal spacing, and others. The ALE method can model solids as well as liquids. The advantage of ALE is the ability to reduce and sometimes eliminate difficulties caused by severe mesh distortions encountered by the Lagrange method, thus allowing a calculation to continue efficiently. However, compared to Lagrange, an additional computational step of rezoning is employed to move the grid and remap the solution onto a new grid [7].

D. SPH

The mesh-free Lagrangian method of space discretization (or SPH method) is a particle-based solver and was initially used in astrophysics. The particles are imbedded in material and they are not only interacting mass points but also interpolation points used to calculate the value of physical variables based on the data from neighboring SPH particles, scaled by a weighting function. Because there is no grid defined, distortion and tangling problems are avoided as well. Compared to the Euler method, material boundaries and interfaces in the SPH are rather well defined and

material separation is naturally handled. Therefore, the SPH solver is ideally suited for certain types of problems with extensive material damage and separation, such as cracking. This type of response often occurs with brittle materials and hypervelocity impacts. However, mesh-free methods, such as SPH, can be less efficient than mesh-based Lagrangian methods with comparable resolution.

Figure 3 gives a short overview of the solver technologies mentioned above. The crucial factor is the grid that causes different outcomes.

The behavior (deflection) of the simple elements is well-known and may be calculated and analyzed using simple equations called shape functions. By applying coupling conditions between the elements at their nodes, the overall stiffness of the structure may be built up and the deflection/distortion of any node – and subsequently of the whole structure – can be calculated approximately [12].

Due to the fact that all engineering simulations are based on geometry to represent the design, the target and all its components are simulated as CAD models [13]. Therefore, several runs are necessary: from modeling to calculation to the evaluation and subsequent improvement of the model (see Figure 4).

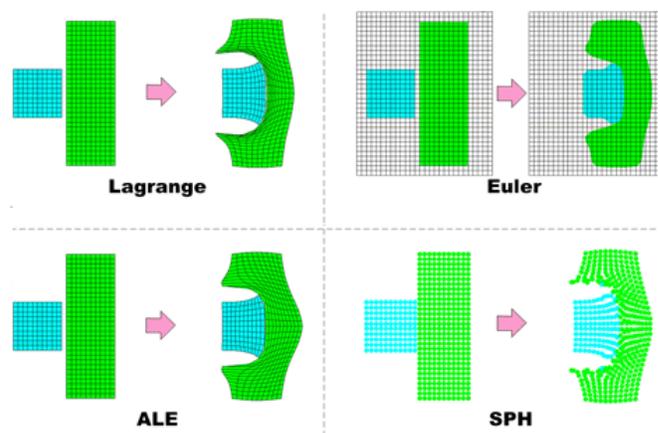


Figure 3. Examples of Lagrange, Euler, ALE, and SPH simulations on an impact problem [11].

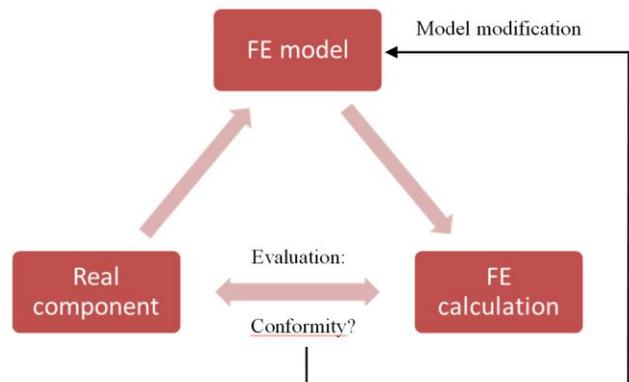


Figure 4. Iterative procedure of a typical FE analysis [12].

The most important steps during an FE analysis are the evaluation and interpretation of the outcomes followed by suitable modifications of the model. For that reason, ballistic trials are necessary to validate the simulation results. They can be used as the basis of an iterative optimization process.

IV. BALLISTIC TRIALS

Ballistics is an essential component for the evaluation of our results. Here, terminal ballistics is the most important sub-field. It describes the interaction of a projectile with its target. Terminal ballistics is relevant for both small and large caliber projectiles. The task is to analyze and evaluate the impact and its various modes of action. This will provide information on the effect of the projectile and the extinction risk.

Given that a projectile strikes a target, compressive waves propagate into both the projectile and the target. Relief waves propagate inward from the lateral free surfaces of the penetrator, cross at the centerline, and generate a high tensile stress. If the impact were normal, we would have a two-dimensional stress state. If the impact were oblique, bending stresses would be generated in the penetrator. When the compressive wave was to reach the free surface of the target, it would rebound as a tensile wave. The target could fracture at this point. The projectile could change direction in case of perforation (usually towards the normal of the target surface). A typical impact response is illustrated in Figure 5.

Because of the differences in target behavior due to the proximity of the distal surface, we must categorize targets into four broad groups. In a semi-infinite target, there is no influence of distal boundary on penetration. A thick target is one in which the boundary influences penetration after the projectile has already travelled some distance into the target. An intermediate thickness target is a target where the boundaries exert influence throughout the impact. Finally, a thin target is one in which stress or deformation gradients are negligible throughout the thickness.

There are several methods which may cause a target to fail when subjected to an impact. The major variables are the target and penetrator material properties, the impact velocity, the projectile shape (especially the ogive), the geometry of the target supporting structure, and the dimensions of the projectile and target.

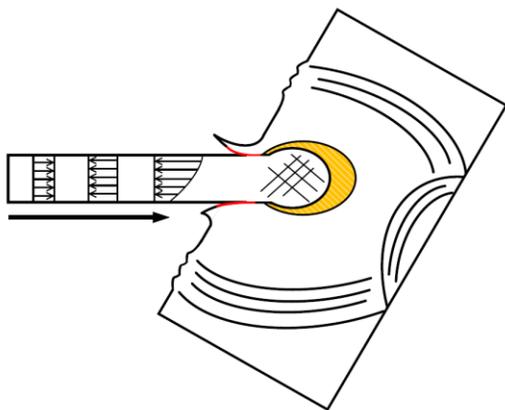


Figure 5. Wave propagation after impact.

The results of the ballistic tests were provided prior to the simulation work to aid calibration. A series of metal plate impact experiments, using several projectile types, have been performed. For the present comparative study, the only target considered is 0.5 inch (12.7 mm) thick 6061-T6 aluminum plate. The plate has a free span area of 8 by 8 inches (203 by 203 mm) and was fixed in place.

The plate was nominally center impacted by a blunt projectile, also made from 6061-T6 aluminum, with an impact speed of 3181 feet/second (970 meters/second). The orientation of the projectile impact was intended to be normal to the target. The projectile is basically a right circular cylinder of length 0.974 inches (24.7 mm) and diameter 0.66 inch (16.7 mm), with a short length of reduced diameter (shoulder) at the rear of the projectile.

The projectile's observed exit speed was 1830 feet/second. The deformed target and projectile are shown in Figures 6 and 7, respectively. As can be seen the target is essentially "drilled out" by the projectile, i.e., a clean hole remains in the target plate. Also, the lack of "petals" on the exit surface of the target indicates the hole was formed by concentrated shear around the perimeter of the hole.

The deformed projectiles, shown in Figure 7, indicate the increasing amount of projectile deformation as it perforates increasingly thicker targets: 0.125 to 0.5 inch. The deformed projectile on the right is the case of present interest. It is worth noting that the simulation of deformable projectiles perforating deformable targets is a challenging class of ballistic simulations. The vast majority of perforation simulations involve nearly rigid projectiles impacting deformable targets. Although deformable projectile calculations form a special, and limited, class in ballistics, establishing confidence in the simulation of this challenging class of problems will lend further confidence to the comparatively easier simulation of near rigid projectile perforating deformable targets [14].



Figure 6. Front view of a perforated aluminum 0.5 inch thick target.



Figure 7. Deformed 6061-T6 aluminum projectiles after perforation 0.125, 0.25, and 0.5 inch thick (left-to- right) aluminum targets.

V. NUMERICAL SIMULATION

The ballistic tests are followed by computational modeling of the experimental set-up.

Three mesh refinement models were constructed using the two-dimension axisymmetric solver in ANSYS. While the three-dimensional solver could also be used, using the two-dimension axisymmetric solver allows more efficient solutions, especially with a large number of elements. The particulars of the three meshes are summarized in Tab. I.

Figure 8 shows two of the three axisymmetric mesh configurations. The mesh discretizations are similar in that each mesh uses one number as the basis for determining the number and size of all the elements in the mesh. The target plate elements immediately below the projectile have the same mesh refinement as the projectile. The configuration is based on [14].

A suite of impact simulations was performed using the above-described 6061-T6 aluminum projectile and 6061-T6 aluminum target. The projectile was given an initial velocity of 3181 feet/second (970 meters/second) and the projectile's speed was recorded at a point near the rear of the projectile. The resulting residual speed was thought to best correspond to the experimental measurement technique for residual speed.

The overall projectile and target plate dimensions were previously given in the description of the ballistic experiment. The axisymmetric model is fully constrained around the outer diameter of the target plate, i.e., fully fixed (clamped).

Different solver methodologies have been applied. The comparison is presented in the following section.

TABLE I. SUMMARY OF MESH CONFIGURATIONS

	Smallest Element (mm)	Number of Elements
Coarse	0.4445	3,174
Medium	0.22225	12,913
Fine	0.14816	28,922

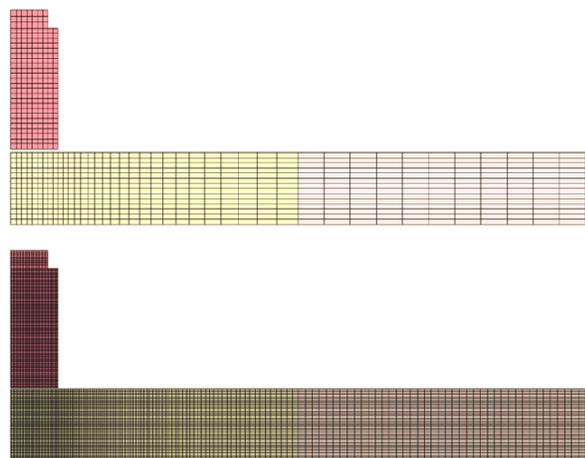


Figure 8. Two of the three axisymmetric mesh discretizations.

A. Solver Evaluation

Using the Johnson-Cook failure criterion eliminates the need to select an erosion criterion and a value for the criterion at which to erode elements. These are two significant difficulties most often overlooked when using an erosion-based simulation technique. Many users select an ad hoc erosion criterion and assign ad hoc values for erosion. In so doing, they seem to ignore the fact that the results are then also ad hoc, which is not desirable when making predictive calculations.

As mentioned above, the Johnson-Cook failure model is not regularized via element characteristic lengths. Thus, we expect the results to be mesh-dependent. It is the purpose of this section to assess this mesh dependency using four successively refined meshes. Subsequently, these Lagrange erosion results will be compared with the corresponding ALE and SPH results.

1) *Lagrange method*: Figure 9 shows the initial and deformed ($t = 0.053$ ms) mesh configurations for the medium discretized mesh. Also shown is an illustration of the eroded element distribution at the end of the simulation. The eroded elements are indicated relative to their initial position using a different color to differentiate them from the non-eroded elements of the same part. Tab. II summarizes the residual speed of the projectile for the three mesh configurations considered. With the exception of the medium mesh speed, which indicates a somewhat larger projectile speed reduction, the projectile speeds are decreasing nearly uniformly with increasing mesh refinement. Figure 10 shows a plot of the residual speed versus the mesh refinement parameter. This plot indicates that the results do not follow the developing trend. Based on this plot, no claim can be made that the results are in the asymptotic regime, much less converged. This is disappointing since the mesh densities for these two cases are likely to be much greater than it would have been attempted in typical three dimensional simulations.

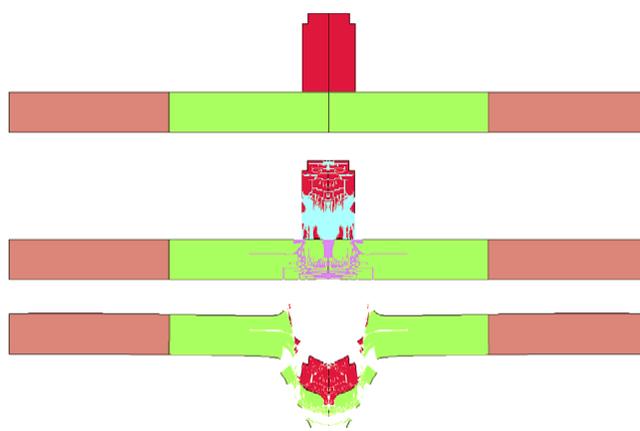


Figure 9. Initial, eroded, and deformed Lagrange elements with a medium mesh configuration.

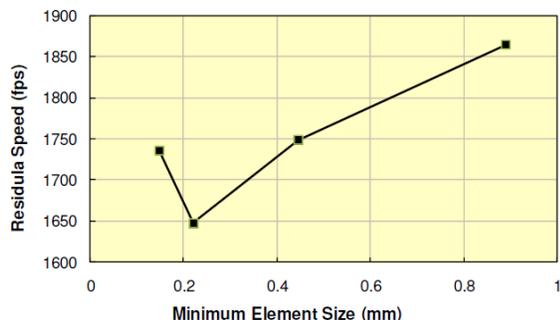


Figure 10. Plot of residual speed versus mesh refinement parameter.

2) *ALE method*: As mentioned above, failure criteria such as the Johnson-Cook failure criterion, cannot be used with Eulerian formulations as cell (element) deletion is not allowed. If a user attempts to use a failure model, the deletion of failed cells will eventually cause the calculation to terminate inaccurately. Thus, all the ALE simulations in this section omit the Johnson-Cook failure model.

In the absence of a failure criterion, it will be demonstrated that the residual speed of the projectile is quite low. It is the purpose of this section to assess the mesh dependency of the ALE solution using successively refined meshes. Subsequently, these results will be compared with the corresponding Lagrange and SPH results.

Note: although the same mesh densities are used in both the Lagrange and ALE simulations in this demonstration, ALE mesh densities generally need to be greater than corresponding Lagrange with erosion mesh densities. The advection of materials from cell-to-cell, and especially the assumption of uniform strain-rate increments for all materials occupying a cell, introduces numerical errors to the ALE solution that can only be minimized by increasing the mesh densities. For the present demonstration, it is posited that the Lagrange mesh densities are greater than they would typically be for such a perforation simulation, making the ALE mesh densities probably appear typical in terms of expectations.

Tab. II compares the previous Lagrange with erosion results with the corresponding ALE projectile residual speeds. The vast majority of perforation simulations involve nearly rigid projectiles impacting deformable targets. Figure 11 shows the ALE simulation at $t = 0.1$ ms with a medium number of elements. It is interesting to note that the ALE deformed projectile is quite similar in shape to the deformed projectile after the test.

TABLE II. COMPARISON OF LAGRANGE WITH EROSION AND ALE PROJECTILE RESIDUAL SPEEDS

Mesh	Residual Speed (fps)	
	Lagrange	ALE
Coarse	1748	1693
Medium	1647	1788
Fine	1737	1834
Experiment	1830	

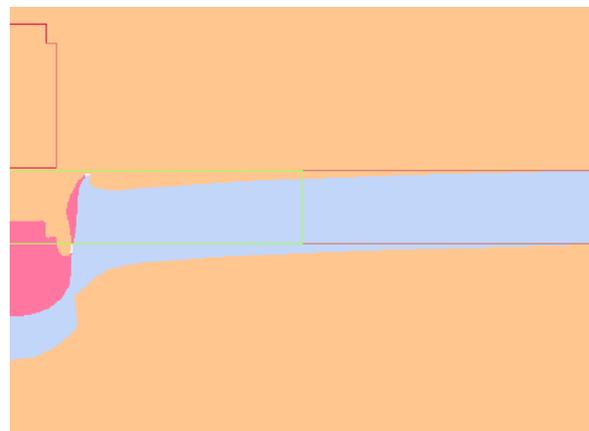


Figure 11. ALE simulation with a medium discretized mesh ($t = 0.1$ ms).

3) *SPH method*: Failure criteria like the Johnson-Cook failure criterion, are not typically used with the Smooth Particle Hydrodynamic (SPH) formulations, as particle methods are designed to avoid mesh distortions, which is the primary motivation for using failure/erosion criteria. It is the purpose of this section to assess the mesh dependency of the SPH solution using three successively refined particle meshes. These results will be compared with the corresponding Lagrange with erosion and ALE results.

Much like the mesh refinements used in the Lagrange and ALE calculations, refinements of the SPH particle spacing requires changing the spacing in both the impacted region of the target plate and the projectile, which is also modeled using SPH particles. Figure 12 shows the coarsest SPH model with the projectile and center of the target plate. It was modeled using SPH particles, while the outer portion of the plate was modeled with Lagrange solid elements. Tab. III summarizes the three SPH meshes.

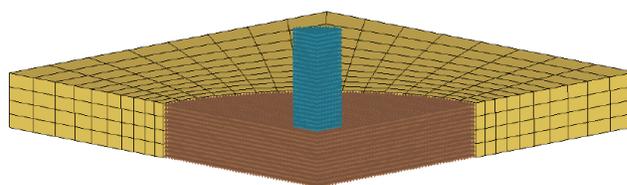


Figure 12. Coarsest SPH model (0.96 mm particle spacing).

TABLE III. SUMMARY OF SPH RESIDUAL SPEEDS FOR THREE PARTICLE MESH REFINEMENTS

Mesh	Particle Spacing (mm)	Number of Particles		Residual Speed (fps)
		Projectile	Target	
Coarse	0.96	1,536	28,665	1094
Medium	0.64	4,860	98,080	1312
Fine	0.43	17,064	333,840	1424
Experiment				1830

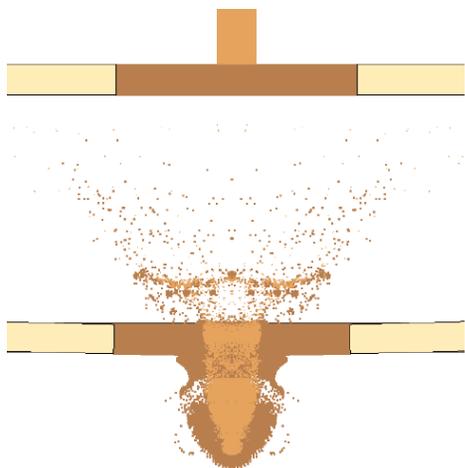


Figure 13. Initial and final ($t = 0.1$ ms) configurations for finest SPH mesh (0.43 mm spacing).

Figure 13 shows the initial and final ($t = 0.1$ ms) deformed projectile and target plate configuration for the finest SPH mesh. In addition to the target plate ‘plug’ being removed from the plate by the projectile (darker brown particles on the right side of target plate), there is considerable front surface ejecta of both the projectile (light brown particles) and the target plate (darker brown particles). For this mesh refinement, the deformed projectile remains relatively intact, with the exception of the front surface ejecta and portions of the projectile that remain attached to the target plate.

B. Simulation Results

In the previous sections, the results from a laboratory experiment were used as a basis to assess the accuracy of the numerical simulations with respect to mesh refinement.

Examining the Lagrange results first without considering the experimental observation, it would seem like the Lagrange method provides the “best” results. All three sets of the Lagrange-with-erosion results have an observed order of convergence which is less than two and thus considered a favorable indication, since few numerical methods have orders of accuracy greater than two.

A general trend seems to be that, as the mesh is refined, the resulting deformed projectile more closely resembles the observed deformed projectile. The exception to this trend is the “point” that protrudes from the front of the projectile. Due to target elements, this “point” appears to be eroded erroneously along the axis of symmetry.

Also, it can be deduced that, for ALE simulations, meshes need to be more dense than it is required for the corresponding Lagrange mesh density. The current status is as follows: the ALE meshes were refined enough, and the Lagrange meshes were more refined than necessary. It is more likely that the advection of material, e.g., from target plate into the surrounding vacuum, over-predicts the motion of the target plate, thus effectively reducing its stiffness and allowing for a “soft catch” of the projectile and an associated reduced projectile residual speed. Here, it needs to be recalled that the Johnson-Cook failure model cannot be

included in the Eulerian simulations as the notion of removal of a cell in the Eulerian context is not permitted. However, these results do indicate that they converge in or at least near the asymptotic range.

Just like the ALE results, the SPH residual speeds increase with increasing mesh density, thus being opposite to the general trend for the Lagrange results. An increasing speed with mesh refinement leads to predictions for a converged result that is greater than the calculated values. Finally, the SPH deformed projectile, previously shown in Figure 13, bears little or no resemblance to the deformed projectile recovered after the perforation test (see Figure 7). Thus, the SPH residual speed results should perhaps be considered reasonable, at least compared to the ALE results. However, the lack of uniformity of the deformed projectile shape between the SPH simulations and the actual experiment might be an indication that the “right” answer might be obtained for the “wrong” reason. Future perforation experiments should include additional diagnostics, e.g., strain measurements on the target plates, so that assessments of agreement can be more extensive than solely considering residual speed.

The SPH deformed projectile looked the least like the observed deformed projectile of any of the three simulation techniques reported. Rather than forming a rounded impact end on the projectile, the SPH deformed projectile seems to form more of a “jet” with a narrow diameter at the fore and a tapered diameter toward the rear. Also, only the refined mesh appears to maintain the integrity of the projectile, i.e., the other two mesh configurations indicate the projectile separating into two parts. Finally, it appears as if some of the projectile material remains on the inner diameter of the hole formed in the target plate. However, it is uncertain if this was observed in the test.

VI. HIGH-PERFORMANCE COMPUTING

The objective is to develop and improve the modern armor used in the security sector. To develop better, smarter constructions requires an analysis of a wider range of parameters. However, there is a simple rule of thumb: the more design iterations that can be simulated, the more optimized is the final product. As a result, a high-performance computing (HPC) solution has to dramatically reduce overall engineering simulation time. HPC adds tremendous value to engineering simulation by enabling the creation of large, high-fidelity models that yield accurate and detailed insights into the performance of a proposed design. HPC also adds value by enabling greater simulation throughput. Using HPC resources, many design variations can be analyzed.

This research will evaluate the performance of the following server generations: HP ProLiant SL390s G7, HP ProLiant DL580 G7, and HP ProLiant DL380p G8.

Taking the influence of the software into account, different versions of ANSYS will be applied here. Regarding the Lagrange solver in a complex 3D multi-material simulation model (modern composite armor structure instead of 6061-T6 aluminum target), the following benchmark is obtained for the different simulations (see Tab. IV below).

The results indicate the importance of high-performance computing in combination with competitive simulation software to solve current problems of the computer-aided engineering sector.

VII. CONCLUSION

The reader is reminded that the ballistic simulation attempted in this work is among the most difficult as both the projectile and target experience significant deformation. The deformation of the projectile as it interacts with the target affects the deformation of the target, and vice versa.

The introduction of a failure criterion, such as the Johnson-Cook failure criterion, is clearly necessary for Lagrange models, and appears to also be necessary for SPH models. A better overall approach than on-off failure models, like the Johnson-Cook failure model, would be the use of continuum damage models. These models allow for the gradual reduction in strength of highly deformed materials and can be used in all three solution techniques.

The importance of mesh refinement has been emphasized in this work. This relatively simple to perform assessment of how the key results change with mesh density is all too often overlooked in computational solid mechanics. Further, establishing that the results are in the asymptotic regime provides some confidence that the mesh density is adequate.

When predictions are required, analysts want as many checks and assurances as possible that their results are credible. Mesh refinement studies provide the analyst some confidence the results are at a minimum not being affected by ad hoc choices of discretization.

This work demonstrates how a small number of well-defined experiments can be used to develop, calibrate, and validate solver technologies used for simulating the impact of projectiles on armor systems.

New concepts and models can be developed and easily tested with the help of modern hydrocodes. The initial design approach of the units and systems has to be as safe and optimal as possible. Therefore, most design concepts are analyzed on the computer.

The gained experience is of prime importance for the development of modern armor. By applying the numerical model, a large number of potential armor schemes can be evaluated and the understanding of the interaction between different materials under ballistic impact can be improved.

The most important steps during an FE analysis are the evaluation and interpretation of the outcomes followed by suitable modifications of the model. For that reason, ballistic trials are necessary to validate the simulation results.

TABLE IV. BENCHMARK TO ILLUSTRATE THE INFLUENCE OF DIFFERENT SERVER AND SOFTWARE GENERATIONS

	ANSYS 14.5	ANSYS 15.0
SL390s G7	27m31s	24m47s
DL580 G7	21m44s	19m51s
DL380p G8	19m16s	14m32s

They are designed to obtain information about

- the velocity and trajectory of the projectile prior to the impact,
- changes in configuration of the projectile and target due to the impact,
- masses, velocities, and trajectories of fragments generated by the impact process.

The combined use of computations, experiments and high-strain-rate material characterization has, in many cases, supplemented the data achievable by experiments alone at considerable savings in both cost and engineering manhours.

REFERENCES

- [1] A. Ramezani and H. Rothe, "Investigation of Solver Technologies for the Simulation of Brittle Materials," The Sixth International Conference on Advances in System Simulation (SIMUL 2014) IARIA, Oct. 2014, pp. 236-242, ISBN 978-61208-371-1
- [2] J. Zukas, "Introduction to Hydrocodes," Elsevier Science, February 2004.
- [3] A. M. S. Hamouda and M. S. J. Hashmi, "Modelling the impact and penetration events of modern engineering materials: Characteristics of computer codes and material models," *Journal of Materials Processing Technology*, vol. 56, Jan. 1996, pp. 847-862.
- [4] D. J. Benson, "Computational methods in Lagrangian and Eulerian hydrocodes," *Computer Methods in Applied Mechanics and Engineering*, vol. 99, Sep. 1992, pp. 235-394, doi: 10.1016/0045-7825(92)90042-I.
- [5] M. Oevermann, S. Gerber, and F. Behrendt, "Euler-Lagrange/DEM simulation of wood gasification in a bubbling fluidized bed reactor," *Particuology*, vol. 7, Aug. 2009, pp. 307-316, doi: 10.1016/j.partic.2009.04.004.
- [6] D. L. Hicks and L. M. Liebrock, "SPH hydrocodes can be stabilized with shape-shifting," *Computers & Mathematics with Applications*, vol. 38, Sep. 1999, pp. 1-16, doi: 10.1016/S0898-1221(99)00210-2.
- [7] X. Quan, N. K. Birnbaum, M. S. Cowler, and B. I. Gerber, "Numerical Simulations of Structural Deformation under Shock and Impact Loads using a Coupled Multi-Solver Approach," 5th Asia-Pacific Conference on Shock and Impact Loads on Structures, Hunan, China, Nov. 2003, pp. 152-161.
- [8] N. V. Bermeo, M. G. Mendoza, and A. G. Castro, "Semantic Representation of CAD Models Based on the IGES Standard," *Computer Science*, vol. 8265, Dec. 2001, pp. 157-168, doi: 10.1007/978-3-642-45114-0_13.
- [9] G. S. Collins, "An Introduction to Hydrocode Modeling," Applied Modelling and Computation Group, Imperial College London, August 2002, unpublished.
- [10] R. F. Stellingwerf and C. A. Wingate, "Impact Modeling with Smooth Particle Hydrodynamics," *International Journal of Impact Engineering*, vol. 14, Sep. 1993, pp. 707-718.
- [11] ANSYS Inc. Available Solution Methods. [Online]. Available from: <http://www.ansys.com/Products/Simulation+Technology/Structural+Analysis/Explicit+Dynamics/Features/Available+Solutions+Methods> [retrieved: August, 2015]
- [12] P. Fröhlich, "FEM Application Basics," Vieweg Verlag, September 2005.
- [13] H. B. Woyand, "FEM with CATIA V5," J. Schlembach Fachverlag, April 2007.
- [14] L. E. Schwer, "Aluminum Plate Perforation: A Comparative Case Study using Lagrange with Erosion, Multi-Material ALE, and Smooth Particle Hydrodynamics," 7th European LS-DYNA Conference, Salzburg, Austria, May. 2009.

Modeling I/O System in HPC: An ABMS Approach

Work-in-Progress Paper

Diego Encinas, Marcelo Naiouf
and Armando De Giusti

Informatics Research Institute LIDI
National University of La Plata
La Plata, Buenos Aires, Argentina
Email: {dencinas, mnaiouf,
degiusti}@lidi.info.unlp.edu.ar

Sandra Méndez

High Performance Systems Division
Leibniz Supercomputing Centre (LRZ)
Garching, Munich, Germany
Email: sandra.mendez@lrz.de

Dolores Rexachs
and Emilio Luque

Computer Architecture
and Operating Systems Department
University Autònoma of Barcelona
Bellaterra, Barcelona, Spain
Email: {dolores.rexachs,
emilio.luque}@uab.es

Abstract—Data-Intensive scientific applications use parallel Input/Output (I/O) software to access files. A tool capable of predicting the behavior of this kind of applications in High Performance Computing (HPC) would be very useful for parallel application developers. On the other hand, Agent-Based Modeling and Simulation (ABMS) has been used to model and simulate problems and complex systems for various science fields. This work presents a preliminary analysis to model a component of the I/O system in HPC using ABMS.

Keywords—Agent-Based Modeling and Simulation (ABMS); HPC-I/O System; Parallel File System.

I. INTRODUCTION

In HPC, increasing demands in I/O system can cause bottlenecks more often. Parallel I/O plays a fundamental role to balance the fast increase in computational power and the progress of processor architectures. Although hierarchical memory architecture helps to mitigate performance penalizations due to disk accesses, its capacity is limited. In critical research areas, like nanotechnology, astrophysics, weather prediction and physical energy, several scientific and engineer simulations are processing more and more data. This fact also reflects the development of new fault-tolerance tools through periodical process state savings (checkpointing). Therefore, it is necessary to identify the factors that affect the performance and to propose new solutions that are capable of reducing performance gap between Central Processing Unit (CPU) and I/O.

The I/O system becomes more complex to provide an appropriate performance and to fulfill the requirements of the parallel applications on the HPC systems. Currently, an I/O system is a multilayer system where each layer has its responsibility and a behavior depending on the system workload. This layered structure hinders to observe the system global behavior, identify which layer is a bottleneck and whether it is doing the I/O inefficiently.

Running the same application with different I/O configurations gives the possibility to tune the I/O system according to the application access pattern. However, tuning the I/O system configuration on the production environment, it is only possible in some layers. Therefore, analyzing application requirements before configuring the real system could be an advantage. One way to predict application performance in HPC systems with

different I/O configurations is using modelling and simulation techniques.

In HPC field, several I/O system simulators have been developed for parallel I/O. Simulator Framework for Computer Architectures and Storage Networks (SIMCAN) [1] is aimed at optimization of communications and I/O algorithms. Parallel I/O Simulator of Hierarchical Data (PIOSimHD) [2] was developed to analyze Message Passing Interface-Input/Output (MPI-I/O) performance and cluster hardware configurations. Co-design of Exascale Storage System (CODES) [3] is a framework for evaluating exascale storage system design points. High-Performance Simulator for Hybrid Parallel I/O and Storage Systems (HPIS3) [4] models the application workloads, parallel file system and storage system of the HPC environment. SIMCAN is based on OMNET++ and PIOSimHD is written in Java. CODES and HPIS3 are built on Rensselaer's Optimistic Simulation System [5], a parallel simulation platform. All these simulators are based on Discrete Event Simulation (DES).

ABMS is an approach to modelling complex systems composed of autonomous agents. Agents have behaviours, often described by simple rules, and interactions with other agents, which in turn influence their behaviours. DES and ABMS work mostly in discrete time but DES is used at low to middle abstraction. Compared to DES, in ABMS the behavior is defined at individual level, and the global behavior emerges as a result of many individuals, each following its own behavior rules, living together in some environment and communicating with each other and with the environment. ABMS is also typically easier to maintain because the model refinements normally result in very local, not global changes [6]. ABMS has been used in various fields, such as economics [7], health [8], biology [9] and sensor networks [10].

Due to the complexity of the modern HPC I/O systems, we consider ABMS appropriate to model and simulate it. We propose a model that represents the global behavior of the I/O system through the interaction of the different agents identified in each layer of the system. For tuning the I/O system, the agent attributes can be modified to observe how impact on the global behavior. When a performance problem is observed, this will be evaluated focusing on a layer, its agents and interactions with environment to detect the source of the problem.

In this context, our aim is to define a preliminary model of the HPC I/O system using ABMS. The proposed model will allow to design different configurations of the I/O system to assess their impact on the system performance. By using the model, we will be able to identify possible bottlenecks and predict the adjustments that will maximize overall system performance.

The rest of this paper is organized as follows. Section II presents the proposed methodology. Section III briefly describes I/O system in HPC. Section IV addresses the development of the initial model. Finally, Section V presents our conclusions and future work.

II. METHODOLOGY

Methodology applied to develop the model is an iterative spiral process. Each iteration involves 5 phases:

- 1) Collecting information of the general functionality of the I/O system in HPC.
- 2) Modeling I/O components and their interactions.
- 3) Implementation of the Model in a Simulator.
- 4) Validation of the simulator considering the information of the general functionality.
- 5) Identifying those components that affect application performance.

Modeling Agent-Based allow represent I/O components that may be either Hardware modules or Software and to refine or update them more easily. Compute nodes, I/O nodes, storage nodes and the network in a computer cluster exhibit a behavior depending on their roles. The processes and the components that form these systems can be modeled with interactions and attributes. The generated models can be used to model a particular cluster. The models must be adjusted with the characteristics of interest and combine them to represent the system behavior.

In the HPC scenario, modeling the behavior of hardware and software requires an appropriate level of abstraction to avoid a degree of detail that will cause a too complex and/or costly simulation. Therefore, the use of agents to model system components and attributes would provide a useful tool for analyzing the I/O behavior of the applications while each component is refined.

III. COMPONENTS OF THE I/O SYSTEM IN HPC

In this section, we present the results of the first phase of the methodology and section III-A details the Parallel Virtual File System (PVFS) selected for the first proposed model.

The HPC I/O systems have different configurations in each computer cluster. Usually, the parallel I/O system is represented as is shown in Figure 1. In this case, it is organized hierarchically according to the path that data follow in a computer cluster.

The parallel I/O system can be observed from three points of view:

- I/O architecture is related to storage and interconnection network features, data management, storage devices location, buffering/caching and storage node availability [11].

- I/O software stack comprises Scientific Data Library (like HDF5 and NetCDF), I/O Middleware (like MPI-IO) and Parallel File System (like GPFS, Lustre and PVFS2) [12].
- I/O access patterns of parallel applications. A parallel application I/O behavior model based on I/O patterns can be used to support the evaluation, design and selection of different I/O configurations [13].

Hardware components can have different roles depending on the data management that is performed by the parallel file system. Therefore, we have selected a parallel file system and analyze its functionality.

I/O libraries can be present in the compute nodes and their configurations affect the performance of the parallel applications.

I/O access patterns of parallel application could obtain different performance depending on the configuration of the I/O libraries, parallel file system or network interconnections. For this reason, the global behavior of the I/O system must consider the software and the configuration of the hardware components.

A. Parallel File System

In HPC, data management is carried out by a parallel file system that supports concurrent access to partitioned and distributed data in the storage system.

PVFS2 is an open source file system developed to support efficient read and write operations of large amounts of data. Due to its modular architecture, it is easy to include new hardware media and new algorithms. This fact demonstrates the PVFS suitability for parallel file system research.

Figure 2 illustrates the modular architecture of PVFS2 [14]. It is designed as a client-server architecture where the server provides the storage and the client contains the access logic to the distributed storage. Servers can be classified into data servers and metadata servers. The formers keep parts of logical files while the latters keep attributes of the logical file system objects (files and directories in PVFS).

IV. PROPOSED MODEL

Our approach for Modeling I/O System in HPC considers:

- I/O components (hardware and/or software) will be individually modelled defining its internal behavior (functional level) and the interactions (interfaces and communication actions) with the rest of the components.
- The functional level allows modelling each one of the specific components through the use of:
 - “State variables” characterizing the main “features” of the specific module/component.
 - “State transition maps” modelling, in a simple and powerful way, the dynamic internal behavior of each component based on the different interactions with other modules guided by the communication actions.

In this work, we begin modeling the I/O system for the use case request attributes of a file to the parallel file system. For example, a parallel application to execute an I/O operation

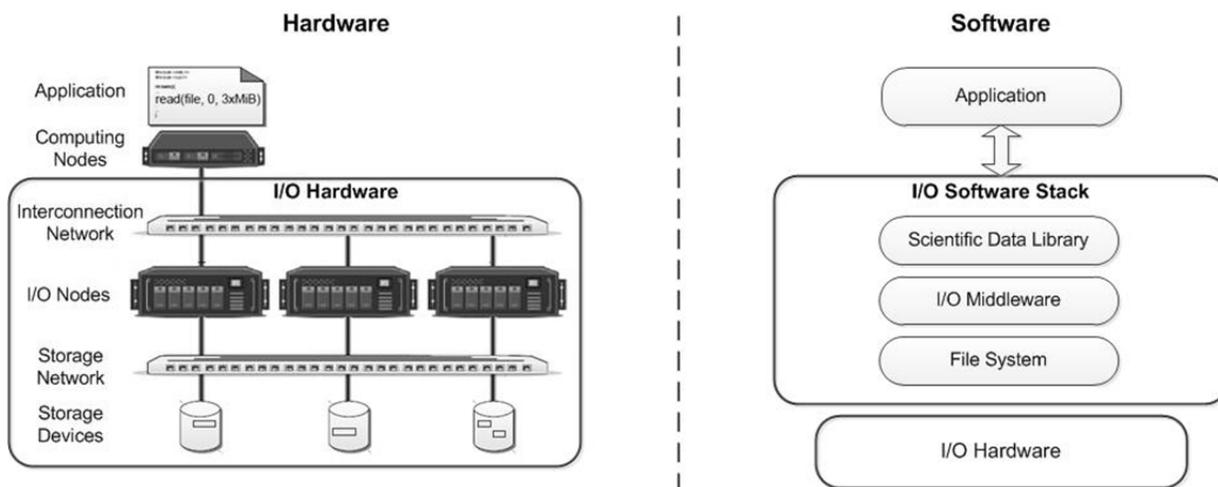


Figure 1. Components of I/O system in a computer cluster. The left picture shows the I/O architecture and the right picture depicts the I/O software stack.

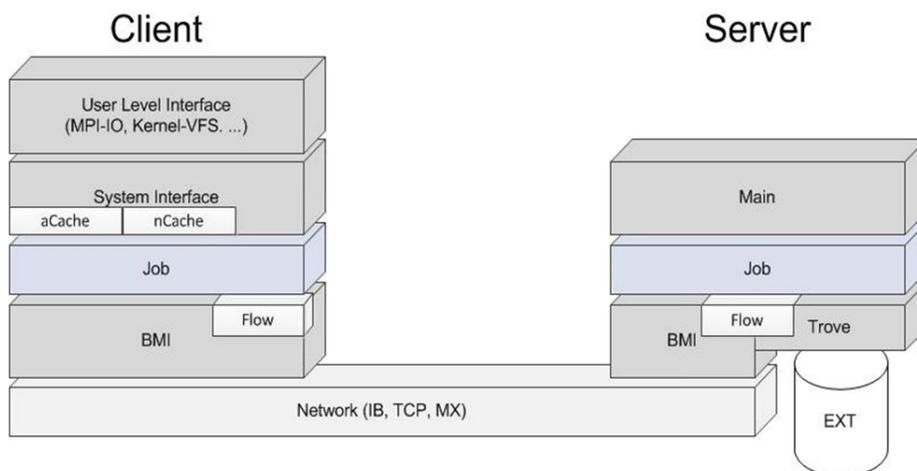


Figure 2. Software architecture of PVFS-2.

(MPI_File_write) will generally need request attributes. For this operation, in Figure 3(a) are shown the involved hardware and software components. The agents Application, Client, Network, Server are identified at high abstraction level. These agents are present in all the I/O systems. The agent Application starts the operation through the Client agent which communicates with the network interface to request attributes to the agent I/O server. The environment for the agents Application and Client is represented by the compute nodes, for the agent I/O server by I/O nodes and storage nodes; and for the agent network interface by the compute nodes, storage nodes and I/O nodes.

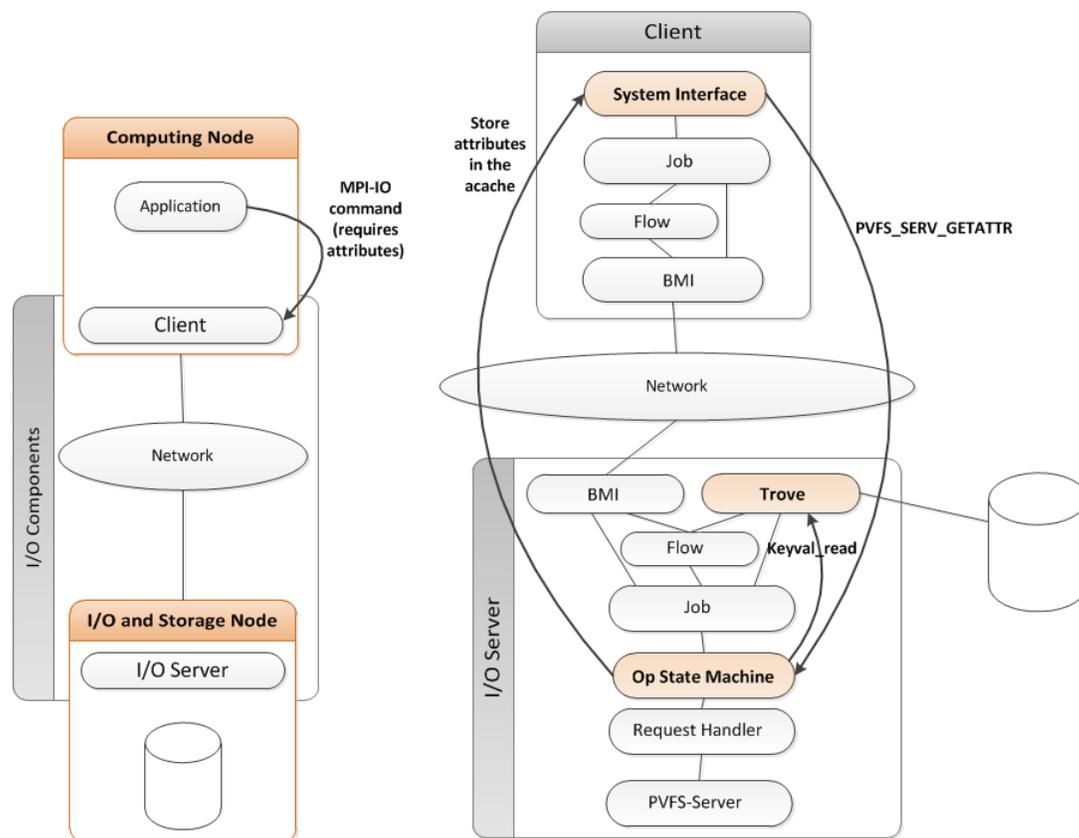
The main components for an I/O system that use PVFS2 is shown in Figure 3(b). The interactions of the agents Client, Network and I/O server were only pointed out at low abstraction level. However, data have went through several components that were not pointed out in this design stage, since only control signals were used as restriction of the initial modeling. One of the modelled agents is the Trove component. Trove has been included in the model because it is the only component that communicates with the storage.

TABLE I. TROVE AGENT

Agent	Variables	Concept	Possible values
Trove	Directories	Object attributes	Attributes, entries (handles, keys)
	Metafile	Object metadata	Datafile handle (keyval), metafile distribution
	Datafile	Logical files data	Data

Trove is a software layer that is responsible of providing and managing persistence in the storage space for the implemented objects. To simulate the significant component behavior, a state machine can be used. The agent Trove state is defined by a collection of state variables that have different possible values. Each state will be associated with a specific combination of state variable values.

Table I shows an initial set of state variables that are defined with basic information to model Trove [15]. This component is selected in first place because it is interesting to model the disk write operation (HD or SSD) in the server node and Trove is the interface that manages it.



(a) Preliminary agents of I/O System in a request attribute operation. (b) Components in an operation to request attributes and preliminary agents in PVFS-2.

Figure 3. Data flow for requesting attributes in PVFS2

In this example, the interactions needed to move from one state to another will be defined by the agent Trove inputs and outputs that correspond to categories provided by Trove interface [16]. These interactions include the management of storage, collections and dataspace and the accesses to Key/values and bytestream.

V. CONCLUSION

This paper presents a preliminary analysis to model a software component of the I/O system in HPC using ABMS. It has been shown by means an example that it is feasible to model PVFS-2 components, its state variables and the interactions with the different agents.

As future work, the agents required to complete the PVFS-2 model will be defined. Next, the model will be extended to include components of the I/O system. In addition, a simulation tool will be defined to implement the simulation model and thus, to be able to reproduce the behavior of the entire system.

ACKNOWLEDGMENT

This research has been supported by the MINECO (MICINN) Spain under contract TIN2011-24384 and TIN2014-53172-P.

REFERENCES

- [1] A. Núñez, J. Fernández, J. D. García, F. García, and J. Carretero, "New techniques for simulating high performance mpi applications on large storage networks," *J. Supercomput.*, vol. 51, no. 1, Jan. 2010, pp. 40–57.
- [2] J. Kunkel, "Using Simulation to Validate Performance of MPI(-IO) Implementations," in *Supercomputing*, ser. Lecture Notes in Computer Science, J. M. Kunkel, T. Ludwig, and H. W. Meuer, Eds., no. 7905. Berlin, Heidelberg: Springer, 06 2013, pp. 181–195.
- [3] N. Liu et al., "Modeling a leadership-scale storage system," in *PPAM (1)*, ser. Lecture Notes in Computer Science, R. Wyrzykowski, J. Dongarra, K. Karczewski, and J. Wasniewski, Eds., vol. 7203. Springer, 2011, pp. 10–19.
- [4] B. Feng, N. Liu, S. He, and X.-H. Sun, "HPIS3: Towards a High-performance Simulator for Hybrid Parallel I/O and Storage Systems," in *Proceedings of the 9th Parallel Data Storage Workshop*, ser. PDSW '14. Piscataway, NJ, USA: IEEE Press, 2014, pp. 37–42.
- [5] C. Carothers, D. Bauer, and S. Pearce, "ROSS: a high-performance, low memory, modular time warp system," in *Parallel and Distributed Simulation, 2000. PADS 2000. Proceedings. Fourteenth Workshop on*, 2000, pp. 53–60.
- [6] A. Borshchev and A. Filippov, "From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools," in *The 22nd International Conference of the System Dynamics Society*, 2004.
- [7] L. Tesfatsion, "Introduction to the special issue on agent-based computational economics," Iowa State University, Department of Economics, Staff General Research Papers, 2001.
- [8] M. Taboada, E. Cabrera, F. Epelde, M. Iglesias, and E. Luque, "Using an agent-based simulation for predicting the effects of patients derivation policies in emergency departments," *Procedia Computer Science*, vol. 18, 2013, pp. 641 – 650.
- [9] A. Siddiqua et al., "A new hybrid agent-based modeling & simulation decision support system for breast cancer data analysis," in *Information and Communication Technologies*, 2009. ICICT '09. International

Conference on, Aug 2009, pp. 134–139.

- [10] M. Niazi and A. Hussain, “Agent-based tools for modeling and simulation of self-organization in peer-to-peer, ad hoc, and other complex networks,” *Communications Magazine, IEEE*, vol. 47, no. 3, March 2009, pp. 166–173.
- [11] D. Kotz, “Introduction to multiprocessor I/O architecture,” in *Input/Output in Parallel and Distributed Computer Systems*, ser. The Kluwer International Series in Engineering and Computer Science, R. Jain, J. Werth, and J. C. Browne, Eds. Kluwer Academic Publishers, 1996, vol. 362, ch. 4, pp. 97–123.
- [12] R. Ross, R. Thakur, and A. Choudhary, “Achievements and challenges for I/O in computational science,” *Journal of Physics: Conference Series*, vol. 16, no. 1, 2005, pp. 501+.
- [13] S. Méndez, D. Rexachs, and E. Luque, “Modeling parallel scientific applications through their input/output phases,” in *CLUSTER Workshops*. IEEE, 2012, pp. 7–15.
- [14] W. Ligon and B. Wilson, “Orangefs,” in *High Performance Parallel I/O*, ser. Chapman & Hall/CRC Computational Science, Q. K. Prabhat, Ed. Chapman and Hall/CRC, 2014, ch. 10, pp. 119–134.
- [15] T. PVFS2, “Trove and PVFS2,” PVFS Development Team, Tech. Rep., 2015.
- [16] S. Lang, “Trove: The PVFS2 Storage Interface,” PVFS Development Team, Tech. Rep., 2015.

Using a Generic Modular Mapping Framework for Simulation Model Composition

Philipp Helle and Wladimir Schamai

Airbus Group Innovations

Hamburg, Germany

Email: {philipp.helle,wladimir.schamai}@airbus.com

Abstract—This paper presents a new framework for solving different kinds of mapping problems, the **Generic Modular Mapping Framework (GEMMA)**. It is geared towards high flexibility for dealing with a large number of different challenges. To this end it has an open architecture that allows the inclusion of application-specific code and provides a generic rule-based mapping engine that allows users without programming knowledge to define their own mapping rules. The paper provides a detailed description of the concepts inherent in the framework and its current architecture. Additionally, the evaluation of the framework in two different application cases, simulation model composition and testbench setup, is described.

Keywords—Mapping; Framework; Simulation Model Composition.

I. INTRODUCTION

Recently, several of our research challenges could be reduced to a common core question: How can we match data from one or more data sources to other data from the same and/or different data sources in a flexible and efficient manner? A search for an existing tool that satisfied our application requirements did not yield any results. This sparked the idea of a new common generic framework for data mapping. The goal in designing this framework was to create an extensible and user-configurable tool that would allow a user to define the rules for mapping data without the necessity for programming knowledge and that yet still has the possibility to include application-specific code to adapt to the needs of a concrete application.

The results of our efforts so far and a first evaluation based on our existing research challenges are presented in this paper.

This paper is structured as follows: Section II provides information regarding related. Section III provides a detailed description of the framework, its core concepts and its architecture. Next, Section IV describes the application cases that have been used to drive and evaluate the framework so far. Finally, Section V concludes the paper.

II. RELATED WORK

The related work divided into two major categories: on the one hand, record linkage and data deduplication tools and frameworks, and on the other hand semantic matching frameworks for ontologies.

Record linkage as established by Dunn in his seminal paper [1] and formalised by Fellegi and Sunter [2] deals with the challenge to identify data points that correspond with each other in large data sets. Typically, this involves databases of different origin and the question, which of the data on one side essentially are the same on the other side even if their name does not match precisely. The same approach is also called data deduplication [3] where the goal is to identify and remove

redundancies in separate data sets. An overview of existing tools and frameworks can be found in [4]. The research work in that area focuses on efficient algorithms for approximate and fuzzy string matching since the size of the data sets involved often leads to an explosion of the run times. These tools [5] often include phonetic similarity metrics or analysis based on common typing errors, i.e., analysis based on the language of the input data. Furthermore, they concentrate on the matching of the string identifiers whereas our framework is more open and flexible in that regard and also includes the possibility to base the matching on available semantic meta-information. The goal in record linkage is always finding data points in different sets representing the same real-world object. Our framework was developed with the goal to match data from different sources that is somehow related but not necessarily referencing the same object.

Semantic matching is a type of ontology matching technique that relies on semantic information encoded in ontologies to identify nodes that are semantically related [6]. They are mostly developed and used in the context of the semantic web [7], where the challenge is to import data from different heterogeneous sources into a common data model. The biggest restriction to their application is that these tools and frameworks rely on the availability of meta-information in the form of ontologies, i.e., formal representations of concepts within a domain and the relationships between those concepts. While our framework can include semantic information, as shown in Section IV-A, it is not a fixed prerequisite.

In conclusion, we can say that our framework tries to fit into a middle ground between record linkage and semantic matching. We use methods applied in both areas but we leave the user the flexibility to choose, which of the features are actually needed in a mapping project.

III. GENERIC MODULAR MAPPING FRAMEWORK

The Generic Modular Mapping Framework (GEMMA) is designed to be a flexible multi-purpose tool for any problem that requires matching data points to each other. The following subsections will introduce the artefacts that make up the core idea behind GEMMA, describe the kind of mapping rules that can be implemented, show the generic process for the usage of GEMMA and describe the software architecture and the current GEMMA implementation.

A. Artefacts

GEMMA is centred around a set of core concepts that are depicted by Figure 1. In an effort to increase the flexibility of GEMMA, the core concepts have been defined in an abstract fashion.

The following artefacts are used:

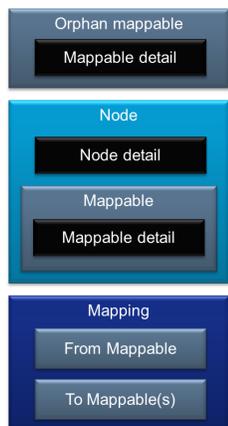


Figure 1. Overview of relevant artefacts

- **Node** - Something that has properties that can be mapped to some other properties.
- **Mappable** - Something that can be mapped to some other thing according to specified mapping rules. Orphan mappables are mappables whose owning node is not known or not relevant to the problem.
- **Mapping** - The result of the application of mapping rules, i.e., a relation between one FROM mappable and one or more TO mappables. Note that the semantic interpretation of a mapping highly depends on the application scenario.
- **Mapping rule** - A function that specifies how mappings are created, i.e., how one mappable can be related to other mappables.
- **Mappable or node detail** - Additional attribute of a mappable or a node in the form of a {detail name:detail value} pair. Details are optional and can be defined in the context of a specific application scenario.

To illustrate these abstract definitions, Figure 2 provides a simple example, where real-world objects depicted on the left hand side are represented on the right hand side in the form of our GEMMA concepts.

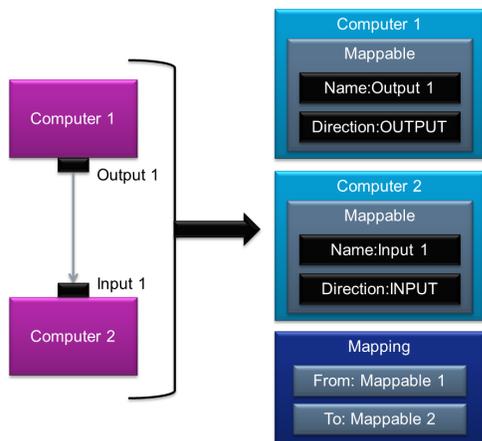


Figure 2. Simple example

In this context, the abstract concept definitions provided above are interpreted as follows:

- **Node** - A computer with input and output ports
- **Mappable** - An input or output port of a computer
- **Mapping** - The connection between ports
- **Mapping rule** - Output ports must be connected to input ports according to some specified criteria such as having the same port name or the same data type.
- **Mappable detail** - Every port has a detail called direction, which defines if the port is an input or output port of the computer

B. Mapping rules

One goal of GEMMA is to allow a large degree of freedom regarding the definition of the mapping rules, so that the framework can be used flexibly for very different kind of application scenarios. So far, the following kinds of mapping rules have been identified and are supported by GEMMA:

- **Name matching**, e.g., map a mappable to other mappables with the exact same name.
- **Fuzzy name matching**, or other forms of **approximate string matching** [8], e.g., map a mappable to other mappables with a similar name (similarity can be based on the Levenshtein distance (LD) [9], i.e., "map" can be matched to "mop" if we allow an LD of 1).
- **Details**, e.g., map a mappable with value of detail $X=x$ to other mappables with values of details $Y=y$ and $Z=z$ or more concretely, map a mappable with detail $direction="output"$ to mappables with detail $direction="input"$.
- **Structured rewriting** of search term based on name, details and additional data, e.g., construct a new string based on the properties of a mappable and some given string parts and do a name matching with the new string (e.g., new string = "ABCD::" + \$mappable.detail(DIRECTION) + "::TBD::" + \$mappable.detail(LOCATION) would lead to a search for other mappables with the name "ABCD::Input::TBD::Front").
- **Semantic annotations** such as user-predefined potential mappings (bindings) using mediators as described in [10], e.g., map a mappable whose name is listed as a client of a mediator to all mappables whose name is listed as a provider of the same mediator.

And, of course, any combination of the above mentioned kinds of rules can be used. For example, structured rewriting could also be applied on the target mappables, which would in effect mean defining aliases for every mappable in the mappable database in the context of a rule.

Additionally, it is planned that several rules can be selected and prioritized for a mapping configuration with options for defining their application, e.g., only if the rule with the highest priority does not find any matches then rules with a lower priority are evaluated.

C. Process

The process for the usage of GEMMA is generic for all kinds of applications scenarios and consists of five steps:

- 1) Import
- 2) Pre-processing
- 3) Matching
- 4) Post-processing
- 5) Export

The mapping process is configured using an XML configuration file that defines which parsers, rules, resolvers and exporters (see III-D for a detailed explanation of the terms) will be used in the mapping project. The open character of GEMMA allows implementing different data parsers for importing data, resolvers for post-processing of the mappings and data exporters for exporting data.

Import loads data into the framework. GEMMA provides the interfaces *DataParser*, *MappableSource* and *NodeSource* to anyone who has the need to define a new data parser for an application-specific configuration of GEMMA. All available parsers are registered in an internal parser registry where the Run Configuration can instantiate, configure and run those parsers, which are required by the configuration file. The data will then be stored in the mappable database.

Pre-processing of data involves selection of mappables that will require matching using whitelists and/or blacklists and structured rewriting of, e.g., mappable names based on mappable details as already discussed in Section III-B. Pre-processing will be user-defined in a set of rules in a file that can be edited with a standard text editor and does not require programming knowledge. The set of rules which should be applied in one mapping project will be defined by the configuration.

Matching involves running queries on the mappable database to find suitable matches for each mappable that is selected for mapping. The queries are derived from the mapping rules. A mapping is a one to (potentially) many relation between one mappable and all the matches that were found.

Post-processing or match resolving is an optional step that is highly driven by the specific application as will be shown in Section IV. It potentially requires the interaction with the user to make a selection, e.g., a mapping rule might say that for a mappable only a one-to-one mapping is acceptable but if more than one match was found then the user must decide, which of the found matches should be selected. Post-processing also allows the user to apply the graphical user interface to review and validate the generated mapping results and thereby to check the completeness and correctness of the defined rules and to manipulate mappings manually, e.g., remove a mappable from a mapping if the match was not correct.

Export is also highly application-specific. Exporting involves transformation of the internal data model into an application-specific output file. Similar to the *DataParser* interface, a generic *MappingExporter* interface allows the definition of custom exporters that are registered in a exporter registry where they can be accessed by the run configuration as dictated by the configuration file.

D. Architecture and implementation

As already stated before, the Generic Mapping Framework is designed as a flexible answer to all sorts of mapping problems. This is represented in the architecture of the framework, which is depicted in Figure 3 in a simplified fashion. GEMMA modules can be categorized either as core or as application-specific. Core components are common for all GEMMA usage scenarios whereas the application specific components have to be developed to implement features that are very specific to achieve a certain goal. For example, data parsers are application-specific as applications might need data from different sources whereas the mappable database and query engine is a core component that is shared. Table I provides a brief description of the most important modules in GEMMA and their categorization.

TABLE I. GENERIC MAPPING FRAMEWORK MODULES

Module	Description	Core	Specific
Data Parser	Reads data (nodes and/or mappables) into the internal data model and feeds the mappable database		x
Generic Mapper	Generates mappings between mappables based on rules	x	
GUI	Interface for loading configuration, displaying mappings as well as allowing user-decisions and displaying of data based on resolver	x	
Issue Logger	Logs mapping related issues	x	
Mappable Database	Stores mappables and allows searches	x	
Resolver	Resolves mappings based on application specific semantics		x
Rule Manager	Reads mapping rules and provides rules information to other components	x	
Run Configuration	Holds the configuration that defines which parsers, exporters and rules are used in the current mapping project	x	
Specific Exporter	Exports the internal data model into a specific file format		x
Standard Exporter	Exports the internal data model into an XML file	x	

The implementation of GEMMA was done in Java. As far as possible, open source libraries and frameworks have been used. The choice for the mappable database, for example, fell on Apache Lucene [11]. Lucene is a high-performance, full-featured text search engine library. The choice of Lucene might seem odd because we are not using it for its originally intended purpose, indexing and searching of large text files, but it offers a lot of the search capabilities like fuzzy name matching that we need and is already in a very stable state with a strong record of industrial applications.

IV. EVALUATION

The evaluation so far has been done using two application cases, simulation model composition and testbench setup.

A. Simulation model composition

The description of the application case simulation model composition requires the introduction of the bindings concept as presented in [10]. The purpose of bindings is to capture the minimum set of information required to support model composition by an automated binding or connecting mechanism. For example, for the outputs of a given component, we wish to identify the appropriate inputs of another component to establish a connection.

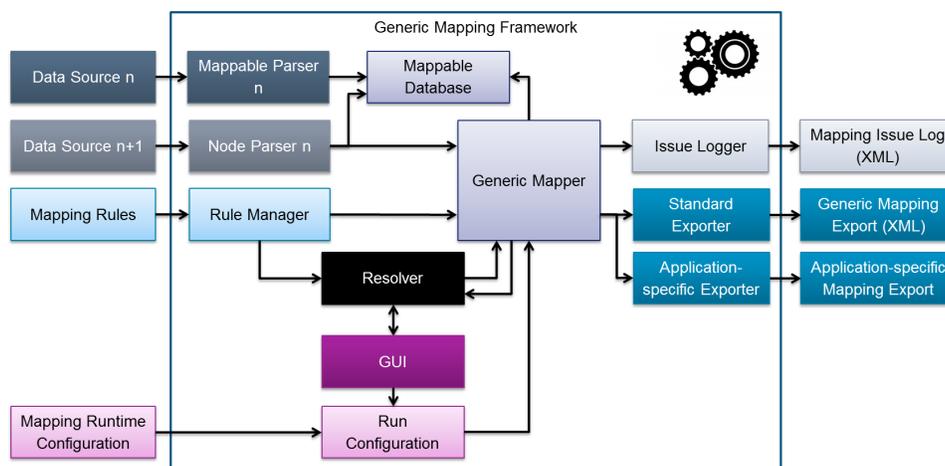


Figure 3. Generic mapping framework architecture

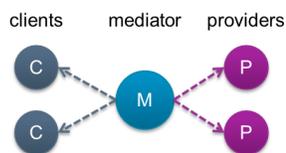


Figure 4. Bindings concept

To this end [10] introduces the notions of clients and providers. Clients require certain data; providers can provide the required data. However, clients and providers do not know each other a priori. Moreover, there may be multiple clients that require the same information. On the other hand, data from several providers may be needed in order to compute data required by one client. This results in a many-to-many relation between clients and providers. In order to associate the clients and the providers to each other the mediator concept is introduced, which is an entity that can relate a number of clients to a number of providers, as illustrated in Figure 4. References to clients and providers are stored in mediators in order to avoid the need for modifying client or provider models.

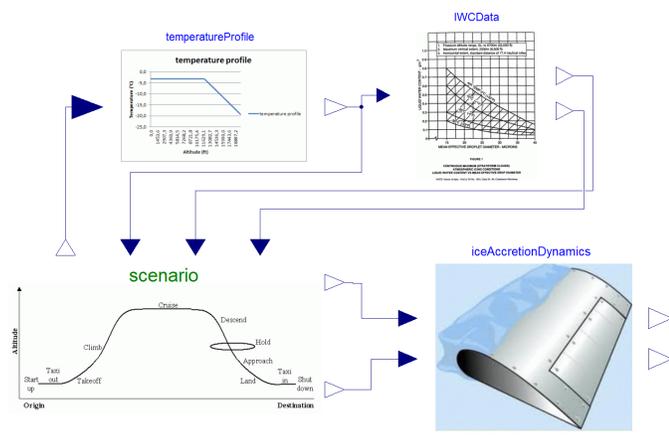


Figure 5. Assembled ice-accretion simulation based on [12]

After the bindings concept was introduced we can now turn to the description of the application case. Generally speaking, the application case is the automatic creation of connections between different model components in a model. Typically in modelling tools, to create a connection between one port of one component to another port of another component requires the user to draw each connection as one line from one port to the other port. If the components' interfaces or the model structure change, then all of the connections have to be checked and some of them have to be redrawn. If we consider a large set of models that have to be changed frequently or if we want to create the models dynamically, then the effort for creating and maintaining the connections between the components in the models becomes a serious issue. The goal of our application case is the formalization of the often implicit rules which the user applies to create the connections to automate this process.

Consider the model depicted by Figure 5, which is a part of the model from the public aerospace use case in the CRYSTAL project [12]. It consists of component models such as flight scenario profile, ice accretion dynamics, and tables for temperature or liquid water content. All of the component models must be interconnected. For example, the temperature profile component requires the current aircraft altitude, which is provided by the flight scenario component; the ice accretions dynamics component requires the current aircraft speed, which is also provided by the scenario component, etc. The individual models were built using the Modelica tool Dymola and exported as Functional Mockup Units [13] (FMUs) in order to be integrated, i.e., instantiated and connected, in a co-simulation environment.

However, assume that the models were created without this specific context in mind. They neither have agreed interfaces, nor do the name and type of the component elements to be connected necessarily match. In order to be able to find the counter parts, i.e., to know that the input of the ice accretion instance should be connected to the appropriate output of the scenario model instance, a dedicated XML file captures some additional information as shown in Figure 6. This way we can capture such interrelations without modifying the models. This data is used as follows: whenever the model "IceAccretionDynamics" is instantiated, bind its input port "speed" to

the output "port p_v", which belongs to the instance of type "ScenarioMissionProfile".

Whenever there will be another model that requires the same data, i.e., current aircraft speed, an additional client entry is added to the same mediator shown in Figure 6. Similarly, whenever there is another model that outputs this data, its corresponding element is referenced in a new provider entry. This approach in particular pays off as soon as there are several models that require or provide the same data. Their connection is then resolved whenever they are instantiated in a specific context model such as the one depicted in Figure 5.

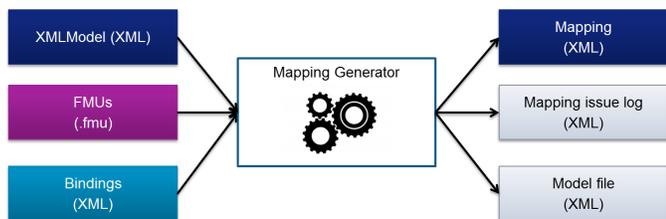


Figure 7. Mapping generator for simulation model composition

In our setting, the bindings specification XML file and the model XML file are application specific sources that are inputs to our generic mapping framework as depicted by Figure 7. The information read from these sources by the application specific parsers is put into the core module mappable database. Two rules, provided by Figures 9 and 10 in pseudocode, are used to query the mappable database to find suitable matches for each mappable. The matching results are then given to the resolver module. This module is aware of the bindings concept and is able to resolve chains of matches and generate a binding for each client and, if necessary, involve the user when an unambiguous mapping is not possible automatically.

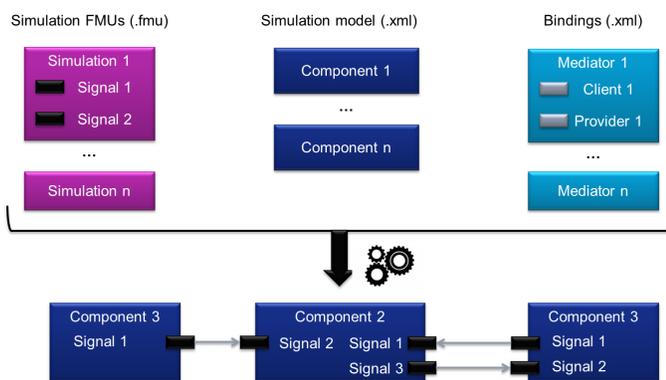


Figure 8. Input and output artefacts of simulation model composition mapper

In the end, the mapping framework uses a list of FMUs, a description of the simulation model consisting of instances of classes implemented in the FMUs and a description of the bindings in the form of an XML file. The output is then the complete simulation model with all the connections between the simulation instances as sketched by Figure 8.

B. Testbench setup

The testbench setup application case is driven by the needs of test engineers. They are given a hardware System under Test

(SuT), a formal definition of the interfaces of the SuT and other equipment and a description of the specified logic of the SuT, which should be tested. Unfortunately, the formal interface definition has been finalized after the specification of the logic, which means that the signal names in the logic description and the signal names in the formal interface definition, which has been implemented in the SuT, do not match. Today, a significant amount of manual effort is required to discover the correct formal signal name for every logical signal. To ease this, GEMMA has been configured as shown by Figure 11.

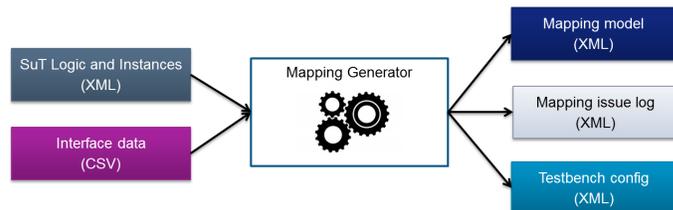


Figure 11. Mapping generator for testbench setup

The goal of the application case is to find a mapping between the name of a signal used in the description of the SuT logic and the corresponding formal interface signal name as shown by Figure 12.

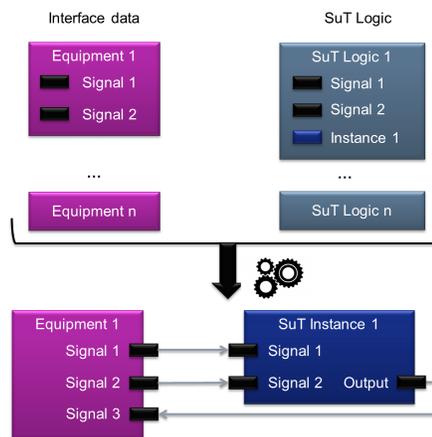


Figure 12. Input and output artefacts of testbench setup mapper

Since the names of the signals could be quite different, the testbench setup application case required the use of the structured rewriting rule type (see Section III-B). One of the rules for the testbench setup is depicted by Figure 13 in pseudo code. The rule defines a new local variable called *soughtName* whose content depends on some attributes of the mappable (enclosed in \$\$) and instead of searching for other mappables that have the same or a similar name as the original mappable, GEMMA searches now for mappables whose name is equal to the variable *soughtName*. If a mappable has the attributes *direction*, *type*, *BLOCKID* and *ID* with the respective values *OUTPUT*, *SuT_Type1*, *45* and *67* then *soughtName* would take the value *AB_BLOCK45_STATUS_67* and GEMMA will search for and map to another mappable in the database with that name.

The main challenge for this application case was the amount of data. Even for a small SuT, the mappable database

```
<mediator name="Current aircraft speed" requiredType="Real">
  <client mandatory="true" qualifiedName="IceAccretionDynamics.aspeed"/>
  <provider qualifiedName="ScenarioMissionProfile1.p_v"/>
</mediator>
```

Figure 6. Example of a binding specification entry in XML format

```
foreach Node:node from source BINDINGS
  foreach Mappable:mappable
    if mappable.detail(CAUSALITY) == INPUT || mappable.detail(CAUSALITY) == PARAMETER
      Search Mappable:otherMappables
        with otherMappables.detail(SOURCE) == BINDINGS
          and otherMappables.detail(CAUSALITY) == CLIENT
          and otherMappables.detail(NAME) == mappable.detail(QUALIFIED_NAME)
        Create Mapping from mapping to otherMappables
      endif
    endfor
  endfor
```

Figure 9. First implemented rule for simulation model composition (in pseudo code)

```
foreach Node:node from source BINDINGS
  foreach Mappable:mappable
    if mappable.detail(TYPE) == PROVIDER
      Search Mappable:otherMappables
        with otherMappables.detail(SOURCE) == XMLMODEL
          and otherMappables.detail(CAUSALITY) == OUTPUT
          and otherMappables.detail(QUALIFIED_NAME) == mappable.detail(NAME)
        Create Mapping from mapping to otherMappables
      endif
    endfor
  endfor
```

Figure 10. Second implemented rule for simulation model composition (in pseudo code)

```
foreach SuTLogicInstance:instance
  foreach Mappable:mappable
    if mappable.detail(DIRECTION) == INPUT
      Search Mappable:otherMappables
        with mappable.detail(NAME) == otherMappable.detail(NAME)
      Create mapping from mappable to otherMappables
    elseif mappable.detail(direction) == OUTPUT
      if instance.detail(type) == SuT_Type1
        soughtName = "AB_BLOCK$$mappable.detail(BLOCKID)$$_STATUS_$$mappable.detail(ID)$$"
      elseif instance.detail(type) == SuT_Type2
        soughtName = "BLOCK$$mappable.detail(BLOCKID)$$_STATUS_$$mappable.detail(ID)$$_CD"
      endif
      Search Mappable:otherMappables
        with mappable.detail(NAME) == soughtName
      Create mapping from mappable to otherMappables
    endif
  endfor
endfor
```

Figure 13. One implemented rule for testbench setup (in pseudo code)

contained 350000 mappables and matches had to be found for 2500 mappables. Nevertheless, the application proved to be successful. The total run time is around 30 seconds including the time for data import and export, and the average time per query is 4.5 ms on a standard PC.

V. CONCLUSION

In this paper, we introduce a new framework for generic mapping problems, GEMMA. It is geared towards high flexibility for dealing with a number of very different challenges. To this end it has an open architecture that allows the inclusion of application-specific code for reading and exporting data and the resolving of mapping results. Furthermore, it provides a generic rule-based mapping engine that allows users without

programming knowledge to define their own mapping rules. So far, the evaluation in the two application cases described in this paper has been highly successful.

As said in Section II, as far as we know, there is currently no other tool with the same functionality as GEMMA. This prevents a direct comparison in terms of performance of GEMMA with other solutions. For our future work we also plan to compare GEMMA functionally to other solutions that rely on more formalized semantic information in the form of ontologies. Depending on the results of this comparison, this might lead to an extension of GEMMA, so that in addition to the matching based on the Lucene text database there will be the possibility to include the results from a semantic reasoner

in the matching process.

From an implementation point of view, there is still a large to-do list:

- Definition of required mapping rules capabilities and format for representing mapping rules.
- Implementation of generic resolver module for post-processing.
- Implementation of generic rule-based mapper.
- Transform project into Eclipse product and use the Eclipse OSGi extension mechanism [14] for registering and instantiating modules.

At the same time, we are actively looking for further application cases to mature the framework.

REFERENCES

- [1] H. L. Dunn, "Record linkage*," *American Journal of Public Health and the Nations Health*, vol. 36, no. 12, 1946, pp. 1412–1416.
- [2] I. P. Fellegi and A. B. Sunter, "A theory for record linkage," *Journal of the American Statistical Association*, vol. 64, no. 328, 1969, pp. 1183–1210.
- [3] Q. He, Z. Li, and X. Zhang, "Data deduplication techniques," in *Future Information Technology and Management Engineering (FITME), 2010 International Conference on*, vol. 1. IEEE, 2010, pp. 430–433.
- [4] H. Koepcke and E. Rahm, "Frameworks for entity matching: A comparison," *Data & Knowledge Engineering*, vol. 69, no. 2, 2010, pp. 197 – 210.
- [5] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: A survey," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 19, no. 1, 2007, pp. 1–16.
- [6] F. Giunchiglia, A. Autayeu, and J. Pane, "S-match: An open source framework for matching lightweight ontologies," *Semantic Web*, vol. 3, no. 3, 2012, pp. 307–317.
- [7] Y. Hooi, M. Hassan, and A. Shariff, "A survey on ontology mapping techniques," in *Advances in Computer Science and its Applications*, ser. *Lecture Notes in Electrical Engineering*, H. Y. Jeong, M. S. Obaidat, N. Y. Yen, and J. J. H. Park, Eds. Springer Berlin Heidelberg, 2014, vol. 279, pp. 829–836.
- [8] P. A. Hall and G. R. Dowling, "Approximate string matching," *ACM computing surveys (CSUR)*, vol. 12, no. 4, 1980, pp. 381–402.
- [9] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8, 1966, pp. 707–710.
- [10] W. Schamai, P. Fritzson, C. J. Paredis, and P. Helle, "ModelicaML value bindings for automated model composition," in *Proceedings of the 2012 Symposium on Theory of Modeling and Simulation-DEVS Integrative M&S Symposium*. Society for Computer Simulation International, 2012, p. 31.
- [11] M. McCandless, E. Hatcher, and O. Gospodnetic, *Lucene in Action*. Manning Publications Co., 2010.
- [12] A. Mitschke et al., "CRYSTAL public aerospace use case Development Report - V2," ARTEMIS EU CRYSTAL project, Tech. Rep. D208.902, 2015.
- [13] T. Blochwitz et al., "The functional mockup interface for tool independent exchange of simulation models," in *8th International Modelica Conference*, Dresden, 2011, pp. 20–22.
- [14] R. Hall, K. Pauls, S. McCulloch, and D. Savage, *OSGi in action: Creating modular applications in Java*. Manning Publications Co., 2011.

Memory Constrained Iterative Phase Retrieval Using GPGPU

Ladislav Mikeš

Institute of Computer Science, Faculty of Science
 Pavol Jozef Šafárik University in Košice
 Košice, Slovakia
 Email: ladislav.mikes@upjs.sk

Abstract—Prediction of a scattering experiment for the new generation of coherent X-ray sources, such as X-Ray Free-Electron Laser (XFEL) requires a significant advance in both precision and numerical effectivity of radiation damage modeling. The preferred method of analyzing data from such experiment to obtain structure information relies on an iterative approach of refining an object estimate by repeatedly simulating the beam propagation between exiting the object and arriving at the detector. Currently, the most cost-effective way is to delegate work to General-Purpose computing on Graphics Processing Units (GPGPU), working around its limitations given by a reduced instruction set, limited memory size and increased latency when accessing the rest of the system. In this paper, we present managing fast GPU memory when performing reconstruction of an object consisting of up to 128 million voxels on currently available devices with less than 6GB of onboard memory.

Keywords—Radiation scattering; Inverse problem; Reconstruction; GPGPU.

I. INTRODUCTION

The emerging of new generation of coherent X-ray sources is characterized by high fluence, femtosecond scale flashes and high repetition rate. After vetoing blank and otherwise defective measurements, we expect a stream of up to 1000 frames per second to be further analyzed. High fluence provides us with enough photons to image a single molecule without need to create crystals, but at the cost of the sample getting destroyed in a rather violent explosion. A detector capable of sampling at 4.5MHz coupled with a very short flash lets us measure the sample when the explosion is just starting and its effects are still negligible [1][2], but to retain detail, we have to consider ongoing radiation damage and integrate over multiple time slices. This requires thousands of the forward scattering problem instances to be solved and considering the projected detector resolution of 1 and 4 MPix, the required computational performance is only attainable by massive parallelism and usage of numerical accelerators, such as GPGPU.

In this contribution, we focus on the partial problem of optimizing limited size single particle reconstruction from low-angle scattering pattern, where only a single node equipped with a single GPU is used. We are going to reconstruct wave phase on detector using the iterative phase retrieval method.

While larger samples can be fixed to a motorized holder to undergo a tomographic scan at smaller doses of radiation, single particles at XFEL are injected into the beam in a stream, losing information about orientation. Common practice is to match the orientation of multiple shots, perform 2D phase retrieval and run a tomographic reconstruction. There are,

however advantages [3] to first generating diffraction intensity volume and then performing 3D phase retrieval. Due to limited GPU onboard memory, performing 3D phase reconstruction at projected detector resolutions with GPGPU is an intricate task and our ongoing research aims to find a reasonable solution.

In Section 2, we describe the general algorithm for phase reconstruction. Section 3 gives an overview of the GPGPU architecture. Section 4 enumerates memory requirements of the basic phase reconstruction algorithm as discussed in Section 2. In Section 5, improvements to address memory consumption concerns are proposed. Results of a test run are shown in Section 6 and conclusions are presented in Section 7.

II. PHASE RECONSTRUCTION

In a scattering experiment, we examine the object interacting with incident radiation. Finding the object corresponding to measured data is more or less straightforward in optical spectrum, but in X-ray, depending on material and wavelength, the imaginary part of refraction index, causing phase shifts in beam, becomes crucial.

While modern detectors are capable of counting X-ray photons at acceptable rates, phase information is lost. However, the phase information can be inferred providing sufficient oversampling [4]. One of the phase problem solutions is the algorithm called iterative phase retrieval [5]. It works on an object estimate and makes gradual improvements to it based on feedback from scattering simulation. To be able to harness the duality of object and its diffraction pattern, we need to understand how a wave evolves between exiting the object and being measured on a detector.

The easiest solution is to place the detector relatively far from our sample and use small angle scattering, which numerically translates to using Fourier transforms, in our case Fast Fourier Transform (FFT). We could also use X-ray optics to further enhance resolution, but this requires more complex propagators and increases reconstruction difficulty.

The general algorithm is as follows:

Start with a flat or a fully random object estimate.

- 1) Propagate beam interacting with object estimate to detector distance
- 2) Update simulated amplitudes to reflect measured data, leave phase unaltered
- 3) Propagate beam back by an inverse operator
- 4) Update object estimate

We iterate these steps over until the object converges and there is nothing to update in step 2.

Since a detector is usually neither a single chip nor a

borderless composite, blind spots are inevitable. Usage can also destroy individual pixels and we will need to ignore them. We therefore keep a pixel validity mask and update only valid ones in step 2.

We usually start with a random object and the best results can be achieved by simply executing the algorithm multiple times and averaging the results.

The object should only take up a portion of reconstruction volume. We employ a support function to differentiate between object and supposedly empty space. The general idea is to bring object estimate inside support closer to backpropagated values while suppressing areas outside support. The support function can be derived from other experiments such as electron microscopy or just estimated roughly, but reconstruction precision and speed require a well-behaved tight support. Ideally, we could derive the support function from object estimate itself by shrinkwrap technique [6], based on a low-pass filter and thresholding. It is numerically more demanding than other steps of one iteration, but we only need to update the support function once every few iterations.

We could also assume object to be positive, since it cannot absorb more radiation than there is and usually does not spontaneously emit radiation either. This, however, can be counter-productive and leave us unable to find the right solution so we use the positivity constraint in a different form.

There has been significant progress made to improve the step 4 [7] and we are currently using the Relaxed Averaged Alternating Reflections method [8]:

$$object_{n+1} = object_n^P + \beta * (object_n - 2 * object_n^P) + \beta * \phi_{S+} (2 * object_n^P - object_n) \quad (1)$$

$$object_n^P = \phi_{BPROP}(\phi_A(\phi_{PROP}(object_n))) \quad (2)$$

Where

$object_n$ is the object estimate in iteration n ,

$object_n^P$ is the object estimate backpropagated from step 3

$\beta \in [0, 1]$ is the update factor

operator ϕ_{S+} applies support and positivity constraint, ignoring (zeroing) negative values

ϕ_A applies measured amplitudes in detector space and

ϕ_{PROP} and ϕ_{BPROP} propagate the exit beam to detector space and back

We further require the information about convergence so we know how much we can trust the reconstructed object. It is extracted as the magnitude of change we had to perform in step 2.

Even though the imaging experiment itself is at most in 2D by design, we are able to run it multiple times for various orientations and put them all together to create a 3D diffraction volume. The same algorithm then applies, but now we use 3D FFT for our propagation instead of its 2D counterpart. For 3D object phase reconstruction we hit memory constraints due to architectural limitations of GPGPU. The ways how to deal with the GPGPU memory constraints are the topic of presented contribution.

III. GPGPU

With increasing demand for 3D features and performance, power of dedicated graphics cards rose rapidly. With higher resolutions, more memory is also required to store color, depth

and stencil buffers, as well as high quality textures and other features.

After the introduction of programmable shaders in the early 2000s, enthusiasts learned to repurpose these to perform various computations on the graphics card, taking advantage of their highly parallelized architecture. The first release of general purpose computing on GPU was with the Nvidia 8800 series, the first CUDA-capable card.

There are 2 major platforms for GPGPU : CUDA by Nvidia and OpenCL by consortium Khronos Group.

- CUDA is available only for chips made by Nvidia and is popular for its easy to use API. Device code (an extension of C) is compiled using a specialized tool and linked into the executable.
- OpenCL was created as an open standard and any GPU manufacturer is welcome to support it. Device code (in C) is loaded, compiled and linked at runtime, making it a somewhat more difficult to manage.

A. Construction

A GPGPU device usually consists of a controller, onboard memory and compute units: cuda cores or stream processing units. A compute unit contains a number of computation cores with their private registers and a small amount of locally shared memory. A program is executed on a block of cores, all performing the same instruction.

Massive parallelism is the main advantage of GPGPU. While individually not as powerful as CPU cores, the sheer number of cores available within a single device makes GPGPU the ideal choice for element-wise operations, a major part of phase retrieval algorithm.

When considering GPGPU for scientific research, it is also important to note their double precision (DP) performance and error correction in memory. These features are exclusive to professional grade cards with double precision performance being reduced to half or even eighth of their single precision (SP) performance as opposed to up to 1/32 in consumer gaming cards. In our case double precision also doubles the memory requirements for most buffers, a price we may not be able to afford in some cases.

B. Memory hierarchy

We can utilize various types of memory on GPU:

- registers are very fast but only in limited supply
- local memory is shared between threads in a block, up to 48kB depending on architecture, 1TB/s throughput
- global memory is shared between everything on device, up to 12GB in current top-end devices, up to 320GB/s
- peer to peer access accessing another device's global memory is limited by speed of their PCIe bus, up to 32GB/s
- system memory limited by PCIe bus and RAM itself, up to 32GB/s

We can see that using memory outside the device is by an order of magnitude slower and is usually used only to load data and save results. Even adding another device with very fast memory of its own may not be a good choice in some cases due to communication bottleneck.

Best practice in cases where memory size is not an issue is to load all input data to GPU memory, perform computation

and save result back to system memory. For problems with localized memory access, we can load only data relevant to current batch being computed and reuse buffers for next batch. There are mechanisms built into current GPGPU architectures to allow for data transfers to/from GPU for next/previous batch while current batch is being computed without significant performance degradation due to concurrent memory access. In our case, we need to have immediate access to all the buffers for various steps of each iteration.

IV. MEMORY USAGE

When performing iterative phase retrieval as discussed above, we need to consider not only the data structures needed to store data, but also temporary storage.

We will express memory requirements using *size* as it mostly depends on reconstructed volume resolution.

A. Basic data structures

As the algorithm suggests, we need place to store:

- input data: *size* floats
- input validity mask: *size* booleans, bytes at best
- object estimate: *size* floats
- object backpropagated: *size* floats
- support function: *size* booleans
- average: *size* floats, [optional]

B. Fast Fourier Transform

There are existing implementations for FFT for both major GPGPU platforms: cuFFT for CUDA and clFFT for OpenCL, both of similar performance. FFT in general requires input and output buffers and temporary storage

- input buffer: *size* complex floats
- output buffer: *size* complex floats
- temporary buffer: *size* complex floats

Commonly used FFT implementations place the zero-frequency element after transforming to the frequency domain at the beginning of buffer, as opposed to the center of image acquired by detectors. While input data can be shifted before some of the other buffers are populated, the shrinkwrap algorithm requires access to zero-centered data or suffers performance losses. Shifting between normal and 0-centered layout requires an additional buffer since it cannot be done with limited register/local memory in-place.

- shift buffer: *size* floats

C. Sequential operations

During thresholding and error calculation, we may need to find the maximum or sum of a buffer. This is seemingly sequential task, but it can be parallelized quite well using the divide-and-conquer strategy.

With built-in mechanisms for synchronization within a thread block we can compute maximum or sum of some part of a buffer. This could be the whole buffer, but then we are limited to using only one thread block and work is done mostly sequentially. What we can do is to have multiple blocks compute over parts of the buffer and when all are finished, we find the maximum or sum of their results. We only need some additional space to hold partial results. Its size does not

depend on reconstruction dimensions, but rather on the number of blocks we wish to use, which is almost negligible.

- divide-and-conquer buffer: *#blocks* floats

D. Shrinkwrap

To perform low-pass filtering, we need a buffer to store temporary result before finding the maximum and applying threshold.

- shrinkwrap buffer: *size* floats

E. Total usage

To sum it all up, we require up to *size**12 floats, *size**2 bytes and another *#blocks* floats. In single precision, this is

$$size * 50 + \#blocks * 4$$

bytes, which for object with 128M voxels (and a negligible number of blocks in comparison) is 6400MB of memory.

The Nvidia TESLA card we were using only had 5.3GB onboard memory, but the memory requirement estimate is upper bound and we can further improve it.

V. IMPROVEMENTS

To fit all the required buffers into available memory, we had to modify the straightforward implementation of iterative phase retrieval and place some limitations on object size.

A. FFT buffers

We can see that the detector space buffer is only used to adjust amplitude. We could keep freeing and reallocating it as needed, but we can also repurpose the space when it is not used.

While CUDA allows us to declare a buffer inside another just by adjusting pointers, in OpenCL we only work with buffer handles and we have to ask for a subbuffer to be created inside another. Another way is to leave one buffer out completely by switching to in-place Fourier transforms.

B. Fourier shift buffer

If reconstruction size is even in all dimensions, shifting is reduced to swapping values in pairs and is possible to do in-place. All we have to do is to pad the input to suitable dimensions, which should be done anyway to ensure good FFT performance.

C. Detector mask

Since the sole purpose of detector mask is to invalidate pixels/voxels, we can do this on the detector data itself. As the amplitude is supposed to be positive, we could use negative values for signify invalid data.

D. Updated total

By doing these simple modifications, we can save *size**3 floats and *size* bytes, reducing total memory usage to

$$size * 37 + \#blocks * 4$$

bytes (SP).

Following mentioned improvements, reconstruction of an object enclosed in a cube with side length of 512 (up to 128M

voxels) now requires 4.74GB of onboard memory for single precision, which fits our device with a safe margin. For double precision it would be 9.34GB. If we wanted to work with data from a 1MPix detector and a cube with 1G voxels, we would need 37GB(SP) or 73GB(DP).

VI. RESULTS

We implemented the algorithm described above to be part of a toolbox called SingFEL as a means of reconstructing objects from experimental data for the Single Particles, clusters and Biomolecules instrument at European XFEL.

The reference implementation on CPU was designed to use matrix element-wise operations, but was only compiled with single-thread support for testing purposes. We compared the runtime for objects of various sizes and the time spent solely on iterations, without I/O or host-device transfers and the GPU version was about 70 times faster.

Shown in Table I is runtime and maximum observed memory usage when running 100 iterations of phase reconstruction. Memory usage increase on CPU can be accredited to Armadillo, linear algebra library, [9] creating temporary buffers for operation results, while in GPU the CUDA runtime requires some additional memory but otherwise fits with our estimates.

TABLE I. PERFORMANCE COMPARISON CPU-1 vs CUDA

Input	CPU	RAM	GPU - CUDA	VRAM
128^3	90s	160MB	1.5s	120MB
256^3	843s	1200MB	12s	620MB
512^3	7020s	9000MB	101s	4810MB

The test was performed on a phantom object (Figure 1)

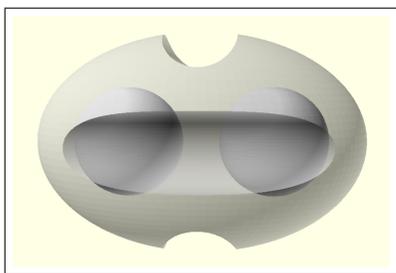


Figure 1. Phantom object

While not representative of our intended target object, a sub-cellular biological sample, it offers well-defined borders and is homogeneous inside for easy error assessment. To simulate a real detector with blind spots, we applied a tic-tac-toe shaped 3D mask to detector data, accounting for missing detector information (Figure 2)

The result after 250 iterations (Figure 3) shows the correct general shape, but is still noisy. We are losing a significant amount of information by masking pixels out.

The error function (Figure 4) approaches zero as there ceases to be any activity during amplitude update.

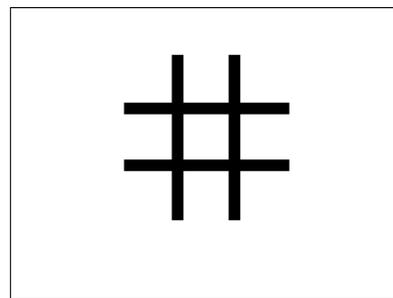


Figure 2. Detector mask, central cut

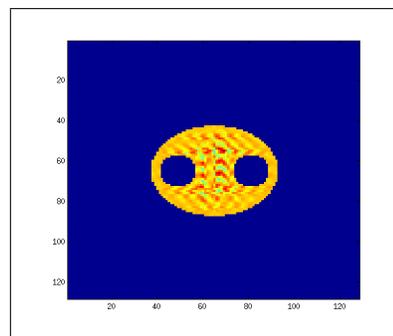


Figure 3. Reconstructed object, frontal cut, off-center

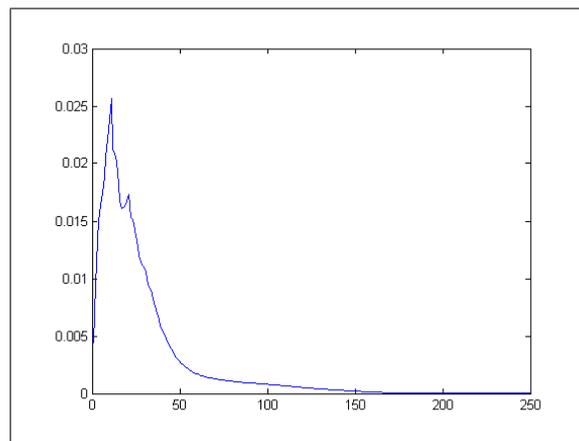


Figure 4. Error measurement

As we can see, the object converges after approximately 170 iterations. If detector data are unmasked, convergence occurs after less than 100 iteration and the difference from input is almost immeasurable in single precision. With real experimental data, the algorithm usually requires hundreds if not thousands of iterations to converge, making the overhead of copying data to GPU insignificant in comparison.

VII. CONCLUSION

We have optimized for GPGPU and implemented the Relaxed Averaged Alternating Reflections method with shrinkwrap-based periodical support update to be included in SingFEL for object reconstruction. We show the best achievable case and why the 1024^3 resolution derived from intended detector resolution is not possible with current hardware and our single GPU limitation. While this is quite certainly a setback, best usable case is still satisfactory for immediate deployment within SingFEL, a software suite for single particle imaging at FEL (in development).

With an average speed-up of 70 over single-threaded CPU implementation, we have also supported the claims about GPGPU being the most cost-effective solution (for reasonably sized problems) to date, as a system with 70+ CPU cores is not going to scale linearly in performance.

At the time of writing this paper, professional single-GPU cards top out at 16GB and dual-GPU cards at 24GB of onboard memory. The first single-GPU card to have 32GB onboard memory was just announced, but still falls short of the estimated 37GB memory required for 1G voxel reconstruction in single precision. Dual-GPU cards may therefore soon surpass this requirement but are still expected to be just as performance limited as multiple cards by having to communicate through the PCIe bus. To push the resolution further, we have to accept a slight bottleneck in inter-device transfers. One of the possibilities would be to dedicate one card to detector (Fourier) tasks. This way, all (2 or 3) FFT buffers and detector data could reside on one card and everything else on the other one. In each iteration we then need to transfer object estimate forward and backpropagated object estimate back, totaling 8GB transfers for 1G voxel reconstruction, delaying each iteration by approximately 1 second at 12GB/s compared to the estimated 8s for a single iteration were it all performed on a single device. We consider this drawback tolerable and will proceed in this direction further.

With double the memory requirements, we did not pay much attention to double precision and the advantages it might bring. Rounding errors usually propagate through iterations, but in case of iterative phase retrieval, the mechanism of replacing amplitudes with measured data creates a fixed point in our computation and error does not compound itself.

On the opposite side, we have the recent introduction of 16-bit floats with the newest CUDA framework. While cuFFT does not yet support half precision, we are considering its feasibility for further development.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Community's Seventh Framework Programme (call FP7-REGPOT-2012-2013-1, proposal 316310) "Fostering Excellence in Multiscale Cell Imaging (CELIM), USP Technicom (ITMS 26220220182) and VVGS-PF-2014-451.

REFERENCES

- [1] R. Neutze, R. Wouts, D. van der Spoel, E. Weckert, and J. Hajdu, "Potential for biomolecular imaging with femtosecond X-ray pulses," *Nature*, vol. 406, no. 6797, Aug. 2000, pp. 752–757.
- [2] Z. Jurek, G. Faigel, and M. Tegze, "Dynamics in a cluster under the influence of intense femtosecond hard X-ray pulses," *The European Physical Journal D - Atomic, Molecular, Optical and Plasma Physics*, vol. 29, no. 2, May 2004, pp. 217–229.
- [3] N. D. Loh and V. Elser, "Reconstruction algorithm for single-particle diffraction imaging experiments," *Physical Review E*, vol. 80, no. 2, Aug. 2009, p. 026705.
- [4] R. H. T. Bates, "Fourier phase problems are uniquely solvable in more than one dimension. I: underlying theory," *Optik*, vol. 61, 1982, pp. 247–262.
- [5] R. W. Gerchberg and W. O. Saxton, "A practical algorithm for the determination of the phase from image and diffraction plane pictures," *Optik*, vol. 35, 1972, pp. 237–246.
- [6] S. Marchesini, H. He, H. N. Chapman, S. P. Hau-Riege, A. Noy, M. R. Howells, U. Weierstall, and J. C. H. Spence, "X-ray image reconstruction from a diffraction pattern alone," *Phys. Rev. B*, vol. 68, Oct 2003, p. 140101(R).
- [7] J. R. Fienup, "Phase retrieval algorithms: a comparison," *Appl. Opt.*, vol. 21, no. 15, Aug. 1982, pp. 2758–2769.
- [8] D. R. Luke, "Relaxed averaged alternating reflections for diffraction imaging," *Inverse Problems*, vol. 21, no. 1, 2005, pp. 37–50.
- [9] C. Sanderson, "Armadillo: An open source c++ linear algebra library for fast prototyping and computationally intensive experiments," Technical Report, 2010.

Resource Management and Optimization Efficiency Issues in a Simulation Software Package

Havatzelet Cohen¹, Daniel Khankin¹, Shlomo Mark¹, Shaul Mordechai²

¹ Shamoon College of Engineering, Beer Sheva 84100, Israel. {havatzelet, daniehe, marks}@sce.ac.il

² Ben-Gurion University, Beer Sheva 84105, Israel. shaulm@bgu.ac.il

Abstract—This work is a component of a larger project that attempts to map the primary factors impacting the software development process of simulation software. In this paper, we describe resource management and optimization efficiency repercussions stemming from the perfective maintenance approach as identified in the development of auxiliary functionality in an advanced Monte Carlo simulation tool that examines the addition of a diamond head attachment to a fiber optic in order to increase the efficiency of the fiber. We found that adding a diamond head did increase the efficiency of the fiber optic, but that modifications necessary to implement the additional functionality in the software required drastically greater computational resources.

Index Terms—*Optimization Efficiency; Perfective Maintenance; Monte Carlo Simulation.*

I. INTRODUCTION

The perfective maintenance of a software product constitutes the modification of the software to improve the effectiveness of the product by adding functionality, as well as improved performance and maintainability. In this paper, we describe the adjust approach for perfective maintenance as expressed in Three-Dimensional Fiber Optic Probe Spectroscopy simulation software (FOPS3D). This software is a complementary scientific software tool that simulates the effects on the absorbance level of the evanescent field of an optical fiber with a tapered thickness. In FOPS3D, we use the Monte Carlo approach by firing rays chosen randomly according to a specified distribution into an optical fiber. The number of collisions encountered within the unclad section are counted and averaged over the (large) number of rays fired into the fiber (histories) in order to yield the average relative absorbance.

In previous work [1][2], the first requirement of the investigation was the development of software that can simulate the effect of tapered fiber thickness on ray absorbance. The Fiber Evanescent Wave Spectroscopy (FEWS) software tool was designed with reference to this requirement. Progress in the research [3][4] required an examination of the effect of fiber bending on the absorbance level and thereby led to new demands for additional functionality permitting user manipulation of the spatial presentation of the tapered fiber. This added capability was a necessary component of research performed for improving the efficiency of the fiber. For this objective, the developer required a change from a combinatorial geometry to a computer graphics approach. A supplementary enhancement

required by the developer, to examine the effects of the attachment of a diamond head to the optical fiber, led us to develop a new version of FOPS3D software that incorporated Diamond Head functionality in other words, a perfective maintenance to the FOPS3D software.

Simulation software development takes place within the domain of scientific software development. At the beginning of the article, we examined the main features of FOPS3D and how it generally relates to scientific simulation software. In this paper, we aim to isolate a particular concern, which is the transformation of a geometrical problem into a graphical one as presented through the model of an optical fiber. Geometry has two primary display options, one of which is combinatorial geometry (CG) with predefined structures that we can use to describe a system.

Using CG as the geometric approach, poses many challenges, one of which is describing complex geometric systems. The most direct way to describe such a system is to use a graphical module to represent the system graphically and to scrutinize the results, as we did with the optical fiber. While, in terms of development, this is a more straightforward approach, it is more vulnerable to the intrinsic limitations in software and hardware. Therefore, the development of scientific software within the strictures of software engineering must take into account many factors, such as efficient use of computing power, software maintainability, Graphics Processing Unit (GPU) rendering capabilities, etc.

In Subsections A of Section II, we provide a detailed physical description of, and exact specifications for, the system being simulated. In Subsections II-B we cover the mathematical constants that were employed for the work. Subsection II-C covers the Monte Carlo approach to the simulation process. In Section III, we discuss the development of the perfective maintenance of the simulation. Finally, in Section IV, we discuss the lessons learned from the process and the course of future development.

II. PHYSICAL BACKGROUND

Previous work [5][6] found that there is a significant difference in the molecular structure of benign lesions spectra as compared with the spectra of malignant cancerous regions. For diagnostic purposes, this discovery led to the application of fiber evanescent wave spectroscopy (FEWS) based on the

phenomenon of attenuated total reflection (Figure 1)) and to a development of a software simulation tool in order to simulate effects on the absorbance of the evanescent field of an optical fiber.

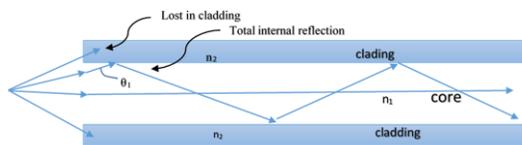


Figure. 1. Ray propagates when $n_1 < n_2$ and $angle > criticalangle$

A. Fiber System Description

The simulation system includes flexible optic fibers, a ray source emitter, and a diamond head in order to improve efficiency. The main properties of the fibers are fiber length, unclad section length, fiber radius, unclad part radius, and reflection coefficients. The simulated Infrared (IR) source emits rays into the fiber. The middle section, which is the region of interest, has its jacket and cladding removed. As shown in Fig. 2, the taper is conical, narrowing down to a thinner cylindrical midsection, with the index of refraction of the core n_1 , the liquid index of refraction n_2 , and n_{cl} of the cladding.



Figure. 2. Tapered midsection

The simulation system allows for user defined properties such as the radius of the simulated fiber, the radius of the unclad part of the fiber, its bending angle, and whether to add a diamond head. Next, the simulated IR source emits rays according to a radial distribution. This is simulated with Monte Carlo techniques. The light beam is propagated through the fiber and hits the fiber medium. The simulation history is defined by the hits of the IR beam in the unclad part that successfully travel through the fiber up to the light sensor. The unbiased mean value of hits is used for calculating the Fresnel transmission coefficient. In the simulation tool, the use of the diamond, increases the number of beam hits in the unclad part by reflecting a significant part of the beam back to the unclad section of the fiber. All of these manipulations can be performed in the simulation, as well as the capability to adjust the radius of the unclad part relative to the rest of the fibers radius. The specifications of the fiber with the diamond head were taken from an existing fiber with a diamond head (Figure 3).As constants in the simulation, we employed a diamond head that is 1 cm long and a fiber with a length of 10 cm.

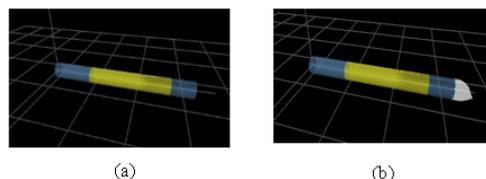


Figure. 3. a) Fiber without Diamond Head. (b) Fiber with Diamond Head.

B. Radiation Source and Absorbance Analysis

In order to describe and sample each incoming ray, we need the radial distance, r , and angle θ , relative to the z axis, when r is a Gaussian distribution for a beam waist as given by [7]:

$$I(r) = I_0 e^{-8r^2/r_0^2} \quad (1)$$

where r_0 is the radius of the laser aperture and I_0 is the irradiance at the center of the aperture. In order for evanescent to propagate the incident angle must be less than the critical angle of the cladded section, the light will be totally reflected without attenuation. The angular distribution given by [8] is:

$$p(\theta) \propto \begin{cases} \sin\theta/\cos^2\theta, & \theta < \theta_{cl} \\ 0, & \theta_{cl} \end{cases} \quad (2)$$

According to the BeerLambert law, the transmitted power $P(z)$ along a fiber is given by [9]:

$$P(z) = P_0 e^{-\gamma z} \quad (3)$$

where P_0 is the power at the input face of the fiber, and γ is the effective evanescent absorption coefficient (which varies for different orientations of the ray) which obey:

$$\gamma(\theta) = \frac{\alpha \gamma n_2 \cos\theta \cot\theta}{2\pi r_1^2 \cos^2\theta_c \sqrt{(\sin^2\theta - \sin^2\theta_c)}} \quad (4)$$

where θ_c is the critical angle of the unclad section $\theta_c = \sin^{-1}\sqrt{n_1/n_2}$ and α is the absorption coefficient of the cladding material at the given wavelength λ given by[10]:

$$\alpha = A e^{-\gamma v/v_0} \quad (5)$$

where v is the optical wave number ($1/\lambda$) and A , γ and v_0 are constants given in data tables [11]. For a simple tube of constant radius, $\gamma(\theta)$ can be computed by integration of (4) over the power distribution of (2). This is very difficult and also inapplicable for a tapered section, where the incidence angles change at each collision with the interfacing surface especially when we use non-point source emitted rays.

C. Monte Carlo Approach

In the Monte Carlo simulation, rays are fired into the fiber according to the radial and angular distributions as per (1) and (2), respectively. Three Cartesian coordinates indicate the ray location and two angles define the direction of the ray movement. In order to define the new location, we define three variables:

- 1) Location, defined by the vector \vec{r}
- 2) Direction of Movement

$$\vec{u} = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)$$

where the angles θ, ϕ are defined relative to the Z axis.

- 3) Distance to the Geometric Boundary ds.

At the geometric boundary, we obtain changes in the direction of the beam and if the ray hits the fiber within the unclad section, the ensuing collisions are counted. The simulation process is based on the realization of the beam movement in a three-dimensional virtual fiber. We employ the Monte Carlo approach to FEWS by firing rays according to a given distribution into an optical fiber with an unclad section. In the simulation software tool, we modeled the absorbance at each point of reflection, and the net absorption - if θ is above the critical angle of the unclad section, the ray is discarded; otherwise, the number of collisions encountered within the unclad section are counted and multiplied by γ/α to provide the absorbance "weight." This figure is averaged (by histories) over a large number of rays fired into the fiber to yield the average relative absorbance.

III. THE PERFECTIVE MAINTENANCE RESULTS

The first software simulation tool was the FEWS application (Figure 4). The FEWS application was built in order to simulate effects on the absorbance of the evanescent field of an optical fiber with tapered thickness. The configuration consists of an IR source that plots rays into a tapered fiber that has had its cladding removed in its middle (tapered) part to allow contact with the sample.

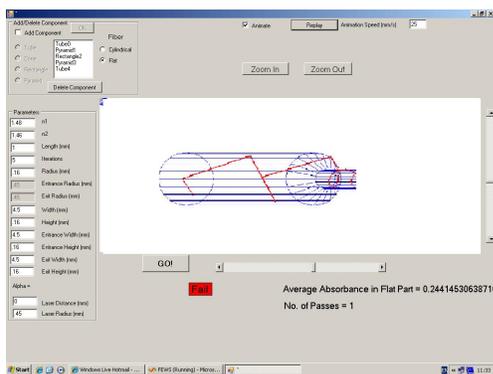


Figure 4. The first software simulation tool

In order to improve the absorbance of the ray, one of the very first suggestions was to taper the fiber by flattening the middle section so that the fiber would have a thinner and flattened midsection (Figure 2). Rachelin [1], who investigated experimentally the hypothesis that changes to a fiber optic, such as the narrowing of its midsection, will increase the absorbance value due to an increased number of reflections, found a correlation between the absorbance and the thickness of the flattened part.

The first simulation tool (FEWS) was built as a prototype in order to reproduce these results by calculating the effect

of the tapered thickness of an optical fiber on the absorbance of the evanescent field of an infrared signal using a Monte Carlo approach. The prototype was found as a suitable [2] tool for analyzing FEWS when the results were in agreement with previous experimental and analytical calculations, that the absorbance increases as the narrowed thickness decreases (Figure 5).

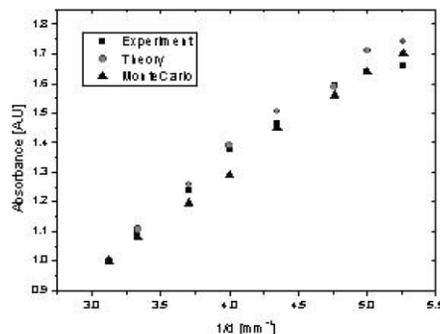


Figure 5. Average Relative Absorbance for Fiber with Narrowed Midsection [2]

As a result, we were required to improve and to extend the capabilities of the tool for fiber with geometries that are more complex as well as to spectroscopy involving more complex physical phenomena.

Progress in the research required continuing development of the software simulation tool. Development demanded changes that focused mainly on the ability to examine the effect of bending the fiber on the absorbance level. This new requirement led to the construction of new software that can cope better with future changes to development requirements. Due to the relatively simple geometry that was being investigated (i.e., a cylindrical fiber with a trapped area), in the early version of the FEWS software, we were able to utilize the well-known geometric description method of CG fundamentals (e.g., bodies and regions).

The new requirements mandated allowing the user to change the spatial presentation of the tapered fiber in order to improve the efficiency of the fiber. These demands obligated the developer to substitute the geometrical description approach with more efficient approach and we choose to use the computer graphics approach. Utilizing the graphical approach, the user determines the geometrical dimensions of the fiber and can also deform the spatial presentation of the fiber. In order to meet these graphical demands, the FOPS3D [4] software tool was built. FOPS3D is a full three-dimensional tool utilizing a computer graphics approach that can simulate any geometrical manipulation of the fiber. The simulation tool was developed in Java language, utilizing the object oriented and event driven concepts of the language. The graphics were programmed using OpenGL. The FOPS3D simulation software product was approved only after passing all testing phases, including a black box testing phase process of validation and verifications against Rachelins [1] results and the FEWS software results [2].

One of the conclusions that was reached during a FOPS3D study of a cylindrical fiber with a bended trapped area (Figure 6) was that the transition from a CG approach to a computerized geometric method resulted in a drastic, but still reasonable, delay in obtaining the results.

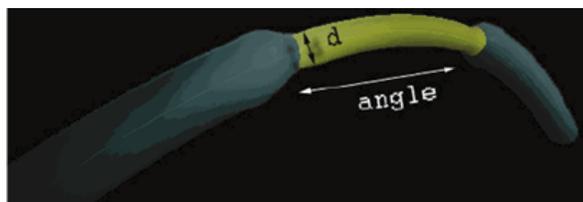


Figure 6. FOPS3D a full three-dimensional tool [4]

As previously described, further research was required in order to improve the efficiency of the optical fiber. Progress in the research led to a focus on improving efficiency without the need for bending the fiber. One possibility was the addition of a diamond head to the fiber tip, based on the theory that a diamond head would increase the absorbance value due to the diamond characteristic of total internal reflection. That is, the diamond would reflect the ray back into the fiber to allow for greater fiber efficiency (Figure 7).

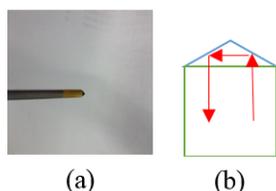


Figure 7. (a) The real fiber used to build the simulated diamond head (b) Ray reflected from the diamond head back to the fiber

The requirement for the use of a diamond head led us to add a new functionality to the FOPS3D, functionality that allowed us to add a diamond head to the fiber. In other words, the new functionality was a perfective maintenance to the FOPS3D software (Figure 8).

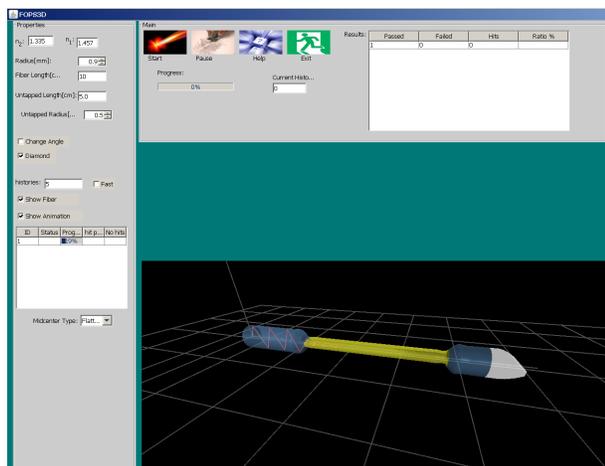


Figure 8. The new version of FOPS3D with the diamond head

The new version of FOPS3D (containing the diamond head functionality), was utilized in the simulation of an un-tapered fiber in order to compare absorbance levels with and without a diamond head. We found that absorbance levels increased by a factor of 2.24 when the diamond head was utilized. In addition, in the case of a tapered fiber with a 0.3 radius, absorbance levels improved by a factor of 1.52, when the diamond head was utilized.

IV. CONCLUSION AND FUTURE WORK

The transition from combinatorial geometry to a computer graphics approach allowed us greater freedom and convenience in depicting the 3D geometric structure of the optical fiber. In developing the first version of FOPS3D, we found that it required a tremendous number of calculations which caused a very significant, but reasonable, delay in obtaining results. This fact did not prevent us from continuing the development of the software. Moreover, in order to reduce the CPU workload and increase performance, we decided to preform these calculations on a GPU.

However, we found that with the addition of the diamond head functionality in the new version of FOPS3D, computing efficiency (measured as the average time to calculate one history) was 2.4 times slower, and above all, required much greater computational resources, such as the extension of RAM memory, the use of a dedicated GPU, and, ultimately, required utilizing computers with multiple processors working in parallel.

In our evaluation of the perfective maintenance process, we found that the required added functionality was successfully implemented. However, from a software engineering standpoint, our criteria were not met, especially because the perfective required considerably greater resources. This determination raised a question about the long-term feasibility of the entire process. In conclusion, extending the functionality of a scientific software product necessarily requires dedicated efforts to achieve the efficient use and management of computational resources (computing power, memory, GPU, processors, etc.), as well as efforts to mitigate the complexity of the algorithmic process. That is, particularly for long-term maintenance, the development process must take into account the corresponding need for improvements to the softwares capacity to contend with the increased complexity of the algorithmic process.

Scientific software product developers must confront a particular challenge at the initial stage of the perfective maintenance process in that, at the outset of the process, there is a circumscribed capacity to prognosticate the full scope, requirements, and efforts (including the necessary computational resources and algorithmic complexity) that will be mandated. Often, until the actual computation is preformed, it is not clear what the most appropriate configuration will be and what will be needed in terms of computing resources, as expressed in this study. Our work is at present continuing in order to improve the simulation process and to optimize the perfective process.

One direction we have taken in our efforts is the modification of the simulation algorithms into distributed versions, such as would allow us to better utilize the resources of emerging cloud technologies.

REFERENCES

- [1] Y. Raichlin, L. Fel, and A. Katzir, "Evanescent-wave infrared spectroscopy with flattened fibers as sensing elements," *Optics letters*, vol. 28, no. 23, pp. 2297–2299, 2003.
- [2] M. Mann, S. Mark, Y. Raichlin, A. Katzir, and S. Mordechai, "Optimization of fiber-optic evanescent wave spectroscopy: a monte carlo approach," *Applied spectroscopy*, vol. 63, no. 9, pp. 1057–1061, 2009.
- [3] D. Khankin, S. Mordechai, and S. Mark, *Monte Carlo Simulation Tool of Evanescent Waves Spectroscopy Fiber-Optic Probe for Medical Applications (FOPS 3D)*. INTECH Open Access Publisher, 2011.
- [4] D. Khankin, S. Mordechai, and S. Mark, "Optimization efficiency of monte carlo simulation tool for evanescent wave spectroscopy fiber-optic probe," *Advances in Optical Technologies*, vol. 2012, 2012.
- [5] D. C. Malins, N. L. Polissar, and S. J. Gunselman, "Tumor progression to the metastatic state involves structural modifications in dna markedly different from those associated with primary tumor formation," *Proceedings of the National Academy of Sciences*, vol. 93, no. 24, pp. 14 047–14 052, 1996.
- [6] Z. Hammody, S. Argov, R. Sahu, E. Cagnano, R. Moreh, and S. Mordechai, "Distinction of malignant melanoma and epidermis using ir micro-spectroscopy and statistical methods," *Analyst*, vol. 133, no. 3, pp. 372–378, 2008.
- [7] E. Hecht, "Optics (international edition)," 2002.
- [8] A. Snider and J. Love, "Optical waveguide theory," *Carman and Hall, New York*, 1983.
- [9] V. Ruddy, B. MacCraith, and J. Murphy, "Evanescent wave absorption spectroscopy using multimode fibers," *Journal of Applied Physics*, vol. 67, no. 10, pp. 6070–6074, 1990.
- [10] T. Deutsch, "Absorption coefficient of infrared laser window materials," *Journal of Physics and Chemistry of Solids*, vol. 34, no. 12, pp. 2091–2104, 1973.
- [11] R. W. Waynant and M. N. Ediger, *Electro-optics handbook*. Citeseer, 2000.

Analytical Modeling of Parametric Yield Considering Variations in Leakage Power and Performance of Nano-Scaled Integrated Circuits

Hossein Aghababa, Nima Mozaffari, and Behjat Forouzandeh

University of Tehran

Tehran, Iran

E-mail: {aghababa}@ut.ac.ir

Abstract— In this paper, we present an analytical method to model the joint probability density function of delay and leakage power. In order to model the joint distribution of these two parameters, they should be modeled independently through an accurate method. The manufacturing process variations as the sources of delay and leakage power variations are considered in our modeling. We also demonstrate that the proposed method is so accurate in modeling joint cumulative density function which is the very parametric yield whose predicting is the main objective of this work. Finally, the proposed method is verified by HSPICE simulations for combinational benchmark circuits in 45 nm technology. We compare the accuracy of our method with that of classic bivariate Gaussian estimation. Simulation results reveal that the mean percentage error of our proposed technique for joint cumulative density function of ISCAS85 benchmark circuits is 2.5 % by average. The average improvement achieved in accuracy of modeling joint cumulative density function through our work compared to aforementioned classic method is 17.1 % and 16.8% respectively without and with considering correlated intra-die variations.

Keywords- *Process Variation; Parametric Yield; Simulation; CMOS Circuits.*

I. INTRODUCTION

The scaling of CMOS has resulted in the increasing magnitude of the variabilities which manifest themselves in the forms of random and systematic variations. Sources of variations could range from random dopant fluctuation to non-idealities imposed by photolithography [1] [2].

The dramatic increase in leakage power with scaling, and a strong dependence of leakage on highly varying process parameters, raises the importance of statistical leakage and parametric-yield optimization. Leakage-power consumption increases are due to both channel (subthreshold)-leakage and gate-leakage currents [3]. The subthreshold current is a strong exponential function of highly variable process parameters. It has been demonstrated that the variability of subthreshold leakage depends primarily on L and V_{th} [4]. Leakage power is inversely correlated with chip frequency. Hence, the Joint Probability Density Function (JPDF) of delay and leakage should be modeled precisely in order to budget one while optimizing another. As a result, parametric yield modeling considering the correlation between delay and leakage consumption is of great importance.

Yield-driven optimization tools require accurate statistical models for both timing and leakage consumption considering correlation between them. There have been works which have

limited their yield estimation to either timing or leakage [5], [6]. However, these approaches neglect the correlation between delay and leakage power.

Although [7] proposes a methodology to model the parametric yield considering leakage/performance correlation, it only presents a closed-form equation for total chip leakage that models the dependence of the leakage current distribution on different process parameters. The only closed-form expression for JPDF of delay and leakage has been proposed by [8][9], wherein the cornerstone of modeling is taking the JPDF of delay and logarithm of leakage as bivariate normal distribution. [10] and [11] despite of improving and building their method upon [8][9] still use the same model which is bivariate normal approximation.

In this paper, we present an analytical approach to accurately model the JPDF of delay and leakage power. In order to verify the efficacy and accuracy of our proposed method, we present the results of HSPICE Monte Carlo simulation as well as those of our model for the performance and leakage consumption of ISCAS85 benchmark circuits [12]. The simulations have been performed in Nangate 45 nm open cell library technology [13]. It should be mentioned that the method presented in this paper is not limited only to this specific issue in Integrated Circuits industry. This method can be extended to any application in engineering and science where modeling and simulation of the JPDF of several random variables is the main objective. This paper is organized as follows. Section II briefly introduces the main formulations we use to model the delay and leakage power under variations. Section III introduces our proposed analytical yield modeling. Section IV discusses the simulation results and a comparison is made between our method and previous work. Finally, Section V concludes the paper.

II. FORMULATIONS OF DELAY AND LEAKAGE POWER UNDER VARIATIONS

In order to model delay and leakage power under process variations, we need to determine which parameters of the circuit are subject to variations. In this paper, similar to [9], we assume that gate-length and gate-length-independent threshold voltage are the sources of variation. It is also assumed that these sources are normally distributed. According to [9], delay and leakage power of an individual gate can be expressed as following:

$$Delay = d_{nom} + \sum_{i=1}^p \alpha_i (\Delta P_i)$$

$$Ln(Leakage) = V_{nom} + \sum_{i=1}^p \beta_i (\Delta P_i) \quad (1)$$

Where d_{nom} and $\exp(V_{nom})$ are the nominal values of delay and leakage power, respectively. α 's and β 's also represent the sensitivities of delay and log of leakage to the process parameters under consideration. The Random Variable (RV) ΔP_i represents the change in the process parameters from their nominal value. In our case, the sources of variations are gate-length and gate-length-independent threshold voltage. Therefore, (1) can be modified to the following for each gate:

$$Delay = d_{nom} + \alpha_1 \Delta V_{th0N} + \alpha_2 \Delta V_{th0P} + \alpha_3 \Delta L$$

$$Ln(Leakage) = V_{nom} + \beta_1 \Delta V_{th0N} + \beta_2 \Delta V_{th0P} + \beta_3 \Delta L \quad (2)$$

where ΔV_{th0N} and ΔV_{th0P} represent the change in NMOS and PMOS gate-length independent threshold voltage. ΔL also represents the change in gate-length.

III. ANALYTICAL YIELD MODELING

The parametric yield of a circuit given delay and power constraints can be expressed as:

$$Y = P(D \leq D0, P \leq P0) \quad (3)$$

which is the probability of delay and leakage being less than $D0$ and $P0$, respectively. Srivastava and Chopra propose a bivariate Gaussian random variable for yield [8][9]. According to [8][9], it is assumed that delay and logarithm of leakage are normally distributed. As a result, they proposed the bivariate normal distribution for JPDP of delay and logarithm of leakage as follows:

$$f_{xy}(x, y) = \frac{e^{-\frac{(x-\mu_x)^2}{\sigma_x^2} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} - \frac{(y-\mu_y)^2}{\sigma_y^2}}}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \quad (4)$$

where, x , y , and ρ respectively stand for delay, natural logarithm of leakage, and correlation coefficient. However, even if the gate delays are assumed to be normal, it should be mentioned that maximum operation is an inherently nonlinear function. The maximum of two normal distributions is a non-Gaussian distribution. Blauuw et al discuss and explain the non-Gaussian nature of delay distribution [14]. Besides, the simulation results reveal that natural logarithm of leakage consumption deviate from normal distribution. For some input states of gates, the leakage distribution does not show lognormal behavior. As a result, the accurate JPDP of delay and natural logarithm of leakage is skewed by both variables. Hence, there is a need to analytically calculate the JPDP of delay and leakage power in order to obtain a more accurate prediction.

A. Analytical JPDP of Delay and Leakage Power

Given two random variables x and y and two functions $g(x, y)$ and $h(x, y)$ we assume we have the following functions:

$$z = g(x, y) \quad , \quad w = h(x, y) \quad (5)$$

These functions are also random variables whose JPDP shall be expressed in terms of JPDP of x and y . Given this fact, we shall express the JPDP $f_{zw}(z, w)$ of the random variables $z=g(x, y)$ and $w=h(x, y)$ in terms of the JPDP $f_{xy}(x, y)$ of the random variables x and y . The Jacobian function $J(x, y)$ is, by definition, the determinant given by:

$$J(x, y) = \begin{vmatrix} \frac{\partial g(x, y)}{\partial x} & \frac{\partial g(x, y)}{\partial y} \\ \frac{\partial h(x, y)}{\partial x} & \frac{\partial h(x, y)}{\partial y} \end{vmatrix} \quad (6)$$

To find $f_{zw}(z, w)$, we solve the systems (5) for x and y . If this system has no real solutions in some region of the zw plane, $f_{zw}(z, w)=0$ for every (z, w) in that region. Suppose, then, that (5) has one or more solutions (x_i, y_i) , that is,

$$z = g(x_i, y_i) \quad , \quad w = h(x_i, y_i) \quad (7)$$

In this case,

$$f_{zw}(z, w) = \frac{f_{xy}(x_1, y_1)}{|J(x_1, y_1)|} + \frac{f_{xy}(x_2, y_2)}{|J(x_2, y_2)|} + \dots + \frac{f_{xy}(x_i, y_i)}{|J(x_i, y_i)|} + \dots \quad (8)$$

where (x_i, y_i) are all pairs satisfying (7) [15]. The aforementioned procedure is extendable to any $n \times n$ systems. Now, for calculating the JPDP of delay and leakage power, we can replace z and w by delay and leakage power, respectively, in previous analysis. However, in our case, we may have n sources of variation where n is not necessarily equal with 2. In this case, we have to add $n-2$ auxiliary equations to systems. Hence, in our case where we have 3 sources of variation, our 3×3 1st order systems are as following:

$$\begin{cases} Delay = z = g(\Delta V_{th0N}, \Delta V_{th0P}, \Delta L) \\ Ln(Leakage) = w = h(\Delta V_{th0N}, \Delta V_{th0P}, \Delta L) \\ E = \Delta L \quad \text{Auxiliary Equation} \end{cases} \quad (9)$$

$$\begin{bmatrix} Delay \\ Ln(Leakage) \\ E \end{bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \Delta V_{th0N} \\ \Delta V_{th0P} \\ \Delta L \end{bmatrix} + \begin{bmatrix} d_{nom} \\ V_{nom} \\ 0 \end{bmatrix}$$

Equation (9) can be simply expressed as follows:

$$Y = \Sigma \bullet X + \Phi \quad (10)$$

Where Y and X are respectively dependent variables and sources of variations vectors. Σ and Φ are respectively $n \times n$ coefficients matrix and constants vectors. Now, (9) should be reordered to solve for vector X . Now, X is obtained from the following:

$$X = \Sigma^{-1} \bullet (Y - \Phi) \quad (11)$$

Therefore,

$$\begin{bmatrix} \Delta V_{th0N} \\ \Delta V_{th0P} \\ \Delta L \end{bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \times \begin{bmatrix} Delay - d_{nom} \\ Ln(Leakage) - V_{nom} \\ E \end{bmatrix} \quad (12)$$

According to (6), (9), the Jacobian function of this system is obtained through the following:

$$J(\Delta V_{th0N}, \Delta V_{th0P}, \Delta L) = \begin{vmatrix} \alpha_1 & \alpha_2 & \alpha_3 \\ \beta_1 & \beta_2 & \beta_3 \\ 0 & 0 & 1 \end{vmatrix} = \alpha_1\beta_2 - \alpha_2\beta_1 \quad (13)$$

Now, by (8) and (13), the JPDF $f_{z,w,E}(z,w,E)$ of delay, Ln(Leakage), and auxiliary variable (E) in terms of the JPDF of ΔV_{th0N} , ΔV_{th0P} , and ΔL can be expressed as following:

$$f_{z,w,E}(z,w,E) = \frac{f_{\Delta V_{th0N}, \Delta V_{th0P}, \Delta L}(\Delta V_{th0N}, \Delta V_{th0P}, \Delta L)}{\alpha_1\beta_2 - \alpha_2\beta_1} \quad (14)$$

Where, ΔV_{th0N} , ΔV_{th0P} , and ΔL are obtained from (12). JPDF of delay and Ln(Leakage) can be obtained through the following integration:

$$f_{z,w}(z,w) = \int_{-\infty}^{+\infty} f_{z,w,E}(z,w,E).dE \quad (15)$$

B. Analytical Parametric Yield

According to (3), the parametric yield of a circuit given delay and leakage power constraints is the probability of delay and leakage power being less than $D0$ and $P0$. Subsequently, having found the JPDF of delay and leakage power, the parametric yield of a circuit can be expressed as following:

$$Y(D0, P0) = \int_{-\infty}^{P0D0} \int_{-\infty}^{D0} f_{z,w}(z,w).dz.dw \quad (16)$$

The parametric yield is a 3-dimensional surface versus

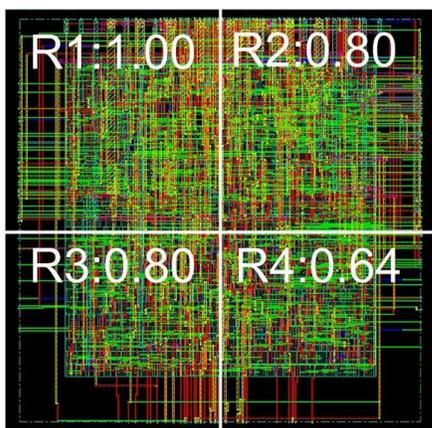


Figure 1. Partition of c7552 physical layout using a grid to model the correlated components of variation where the reference for correlation coefficients is the top-left square.

delay and leakage power constraints. This surface monotonically increases from 0 to 1 as it represents a probability.

C. Correlated Intra-Die Variations

The sources of variation are correlated within the die area. To handle the correlated components of variations, the overall chip is divided into a grid as shown in Figure 1. The correlation coefficient varies from one and falls off to zero with increasing distance. Each square in the grid corresponds to a random variable of the process parameter which has correlations with all other random variables corresponding to other squares in the grid. The values at the top of each of the grids show the correlation coefficients with the top left square on the grid.

To simplify the problem, this set of correlated random variables is replaced by another set of mutually independent random variables with zero mean and unit variance using the principal components of the set of correlated random variables. A vector of random variables, say X , with a correlation matrix C , can be expressed as a linear combination of the principal components Y as [8]:

$$X = \Delta_X + \Omega V^{-1} D^{1/2} Y \quad (17)$$

Where Δ_X is the vector of the mean values of X , Ω is a diagonal matrix with the diagonal elements being the standard deviations of X , V is the matrix of the eigenvectors of C , and D is a diagonal matrix of the eigenvalues of C . If Δ_{P_p} in (1) is generated by (17), correlated sources of variations are incorporated in calculating delay and leakage power consumption.

IV. RESULTS AND DISCUSSION

We implemented our proposed model and bivariate normal model in MATLAB and compared the accuracy of both approaches by HSPICE Monte Carlo simulations. We performed 10,000 Monte Carlo simulations on ISCAS 85 benchmark circuits assuming that all sources of variations have ($3\sigma/\mu=20\%$). In this paper, we assume that sources of variations are channel-length and gate-length independent threshold voltages of NMOS and PMOS transistors. In this work, the full-chip simulations on ISCAS 85 circuits have been set up similar to those of [8] in terms of sources of variations. In our approach, the JPDF of delay and leakage power are calculated analytically by means of Jacobian matrix.

Figure 2 depicts the JPDF and Joint Cumulative Distribution Function (JCDF) of bivariate normal model for c432 from the benchmark circuits. Figure 3 shows the JPDF and JCDF of our model and simulation results. As shown in Figure 2 and Figure 3, the JPDF of our model is visually more similar to that of simulation results compared to bivariate normal model. The errors are calculated with regard to HSPICE simulation results. The Mean Percentage Error (MPE) of model is calculated as following:

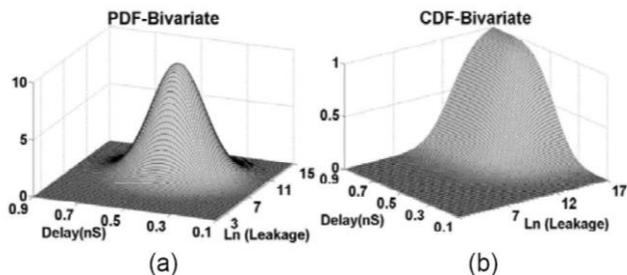


Figure 2. Bivariate normal model diagram of c432 versus Delay and Ln (Leakage) under variations having ($3\sigma/\mu = 20\%$) (a) Joint PDF (b) Joint CDF.

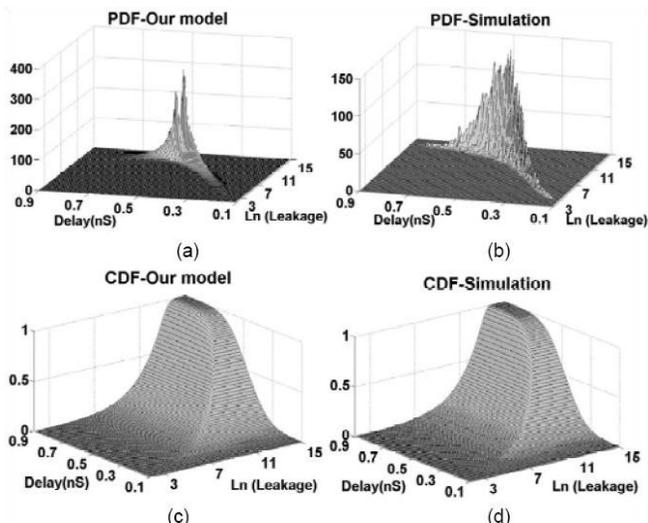


Figure 3. Comparing the results of JPDF and JCDF of our model with those of HSPICE simulation under variations having ($3\sigma/\mu = 20\%$) for c432 (a) JPDF of our model (b) JPDF of simulation (c) JCDF of our model (d) JCDF of simulation.

$$MPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{Model(i) - SPICE(i)}{SPICE(i)} \right| \quad (18)$$

where $Model(i)$ and $SPICE(i)$ are the i^{th} point in the model and SPICE simulation, respectively. The improvement in the mean error achieved by our model compared to bivariate model for ISCAS 85 benchmark circuits are collected in Table I with and without correlated intra-die variation. We have calculated the average error improvement achieved for estimating the JCDF of delay and leakage by our model to be 17.1% compared to bivariate normal model when correlated intra-die variations have not been considered. The mentioned average error improvement has been calculated to be 16.8% by considering correlated intra-die variations. We have assumed that the correlated intra-die variations follow the pattern of Figure 1 for all circuits.

V. CONCLUSION

We proposed a methodology for modeling the JPDF and parametric yield considering variations of delay and leakage power. In this method, we analytically calculated the JPDF of leakage power and delay by means of Jacobian matrix. Then,

TABLE I
MEAN PERCENTAGE ERROR (%) OF JCDF OBTAINED FROM OUR MODEL AND BIVARIATE NORMAL MODEL

Test Gates	Without Correlated Intra-Die Variations		With Correlated Intra-Die Variations	
	Bivariate Model	our Model	Bivariate Model	our Model
c17	21	1.7	19	1.6
c432	15.2	1.3	14.1	1.35
c499	14.4	1.6	15.2	1.67
c1355	20	1.7	21.4	1.65
c1908	19.8	1.4	18	1.25
c2670	26	5.2	24.3	3.98
c3540	22	3.6	21	3.24
c5315	17.8	4.2	19	5.15
c6288	23.4	2.7	22.6	3.1
c7552	16.8	1.3	18.7	1.53
AVG	19.6	2.5	19.3	2.5

JCDF (parametric yield) was extracted by integration. We demonstrated that JPDF and JCDF achieved by our modeling outperform their counterparts in [8][9]. Simulation results in 45 nm technology on ISCAS85 benchmark circuits revealed that our proposed model improves the accuracy of JCDF 17.1% and 16.8% by average, respectively, without and with considering correlated intra-die variations. The proposed method in this paper is generic and can be incorporated in modeling any JPDF in engineering, considering the correlations among the random variables, where accuracy of the model is of great importance.

REFERENCES

- [1] M. Orshansky, S. R. Nassif, and D. Boning, Design for Manufacturability and Statistical Design: A Constructive Approach. New York, NY: Springer, 2008.
- [2] A. Srivastava, D. Sylvester, and D. Blaauw, Statistical Analysis and Optimization for VLSI: Timing and Power. New York, NY: Springer, 2005.
- [3] M. Mani, A. Devgan, M. Orshansky, and Y. Zhan, "A statistical algorithm for power and timing-limited parametric yield optimization of large integrated circuits," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 26, no. 10, Oct. 2007, pp-1790-1802.
- [4] R. Rao, A. Devgan, D. Blaauw, and D. Sylvester, "Parametric yield estimation considering leakage variability," in Proc.Des.Autom.Conf., 2004, pp. 442-447.
- [5] H. Chang and S. S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single PERT-like traversal," in Proc. ACM/IEEE Int. Conf. Comp. Aided-Des. ICCAD, 2003, pp. 621-625.
- [6] C. Viswesweriah et al, "First-order incremental block-based statistical timing analysis," in Proc.Des.Autom.Conf., 2004, pp. 331-336.
- [7] R. R. Rao, A. Devgan, D. Blaauw, and D. Sylvester, "Analytical yield prediction considering leakage/performance correlation," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 25, no. 9, Sep. 2006, pp-1685-1695.

- [8] A. Srivastava et al, "Accurate and efficient gate-level parametric yield estimation considering correlated variations in leakage power and performance," in Proc.Des.Autom.Conf., 2005, pp. 535-540.
- [9] A. Srivastava, K. Chopra, S. Shah, D. Sylvester, and D. Blaauw, "A novel approach to perform gate-level yield analysis and optimization considering correlated variations in power and performance," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 2, Feb. 2008, pp-272-285.
- [10] K. Chopra, S. Shah, A. Srivastava, D. Blaauw, and D. Sylvester, "Parametric yield maximization using gate sizing based on efficient statistical power and delay gradient computation," in Proc. Int.Conf.on Comp.-Aided Design, 2005, pp. 1023-1028.
- [11] D. Sylvester, K. Agarwal, and S. Shah, "Variability in nanometer CMOS: impact, analysis, and minimization," Elsevier the VLSI Journal of Integration, 41 (2008), pp. 319-339.
- [12] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits," Proc. IEEE Int. Symp. Circuits and Systems, IEEE Press, Piscataway, N.J., 1985, pp. 695-698.
- [13] FreePDK45. [online] Available: <http://www.eda.ncsu.edu/wiki/FreePDK45:Contents> [Retrieved, October 2015]
- [14] D. Blaauw, K. Chopra, A. Srivastava, and L. Scheffer, "Statistical timing analysis: from basic principles to state of the art," IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 27, no. 4, Apr. 2008, pp-589-607.
- [15] A. Papoulis, Probability, Random Variables and Stochastic Process. New York: McGraw-Hill, 2002.

Combining Finite Element and Multibody Modeling Techniques for Time-Efficient Simulation of Nonlinear Gear Dynamics

Jakub Korta*, Antonio Palermo^{†*‡}, Domenico Mundo* and Shadi Shweiki*[†]

*University of Calabria, Department of Mechanical, Energy and Management Engineering,
Ponte Pietro Bucci, Cubo 46C, 87036 Rende, Italy

[†]Siemens Industry Software NV, Interleuvenlaan 68, 3001 Leuven, Belgium

[‡]KU Leuven, Department of Mechanical Engineering, Celestijnenlaan 300 B, B-3001, Heverlee, Belgium

Email: jakub.korta@unical.it, antonio.palermo@student.kuleuven.be,
domenico.mundo@unical.it, shadi.shweiki@unical.it

Abstract—Dynamics and vibroacoustics of mechanical transmissions are perceived as one of the major concerns in the contemporary geared driveline design. These characteristics influence durability and efficiency of the system, as well as its quality. However, due to strong local effects and system nonlinearities, computer analyses of gears can be extremely time consuming, and hence, in the industrial practice, are carried out only to some limited extent. This paper describes a time-efficient multibody approach to this problem, which allows for fast analyses of high fidelity, complex numerical gearbox models. The approach presented in the paper consists of two phases: firstly, static transmission error curves, describing the gear meshing stiffness, are derived by analyzing high fidelity Finite Element models of the gears; subsequently, time-efficient multibody simulations of gear dynamics under different operating conditions are achieved by using a specifically defined user force element that allows to take the meshing stiffness variability into account thanks to the previously calculated static transmission curves.

Keywords—Gears; Mechanical Transmissions; Transmission Error; Dynamics; Nonlinear Statics.

I. INTRODUCTION

The design of a drivetrain based on a gearbox requires understanding of a number of different mechanical engineering fields, among which material strength, fatigue, noise and vibration (NV) or manufacturing issues are particularly relevant. The contemporary methods used to support mechanical design process take advantage from numerical tools, which allow to prepare and test a number of alternatives in relatively short time and without the need of preparing expensive physical prototypes. However, in the case of the mechanical transmissions, to obtain reliable and accurate results, it is required to use high fidelity, very detailed computational models. This drives a need for accurate experimental campaigns, which could be used for model validation purposes. A description of a modern gear test rig and a technique for measuring the transmission error between two meshing gears can be found in Ref. [1]. The difficulty in gear numerical modelling is driven by the fact that the phenomena observed on the surface of gear teeth are often local (e.g., Hertzian stress) and nonlinear (e.g., contact force moving along the tooth profile). The former problem is caused by high forces acting on a very small, curved contact surface of the teeth. High, Hertzian stress develops locally and causes local material deformation, which influences the teeth contact mechanics. The nonlinearities instead, can have various sources, among which the most significant are connected with the number of teeth staying in contact during gear rotation (expressed by the so-called contact ratio number

- see Figure 1) and by nonlinear deflection of a tooth, which is caused by contact force traveling along its profile, which normally has uneven thickness from the bottom, to the top. In particular, three-dimensional dynamic analyses aimed at assessing drivetrain NV characteristics are often very time consuming and require significant computational power. To overcome this problem, it is common to describe the analyzed gears by simplified, one-dimensional or planar models [2] [3] [4], which can cover the general behavior of the system, representing roughly its deflection caused by loading forces. In the same time, however, they neglect important phenomena influencing its performance: misalignments, shuttling or teeth microgeometry modifications. These factors contribute considerably to dynamic loading of the gears, causing gear excitation and, as a consequence, system vibrations.

The term of misalignment refers to inaccuracies in gear relative positioning, which result in unsymmetrical load distribution on teeth surfaces. These positional errors can be caused by lack of parallelism between the shaft rotation axes, deflection of the mechanism components or manufacturing errors (e.g., gear or shaft eccentricity). Shuttling, which is a fluctuation in the axial position of the resultant contact force (i.e., along the rotation axis), leads to oscillations on bearing forces and dynamic moments in the plane of action (i.e., the plane spread over the contact force vectors, along the axial direction). Shuttling happens intrinsically for gears with inclined teeth (i.e., helical gears), due to the traveling contact areas, from one corner of the tooth surface to the opposite, when the gears spin. Other reason can be due to shifts in contact stress distribution caused by gear misalignment [5]. Microgeometry modifications (e.g., tip relief, crowning) are applied often to the gears in order to improve the interaction between the teeth and to decrease local stress concentration on the teeth edges [6]. In Ref. [7], the authors analyzed the dynamic characteristics of a gear pair, which was subjected to excitation caused by external forces and geometry inaccuracy. This paper provides a wide theoretical insight into the gear dynamics problem, proposing an analytical approach which can be used to solve it. The authors described also numerical calculations, which were aimed at understanding the influence of forcing parameters on the analyzed system. However, the proposed method does not rely on MB modelling technique, severely limiting its applicability in the industrial practice.

Other factors which should not be neglected when gears are under NV analysis are the variable meshing stiffness and the Transmission Error (TE). The former describes fluctuation

of gear pair stiffness and is caused by uneven load distribution among the teeth during the gear rotational movement (see Figure 1). It is governed by the gear contact ratio and is load dependent. Nonlinear stiffness encountered in mechanical transmissions was studied by using a piecewise representation in Ref. [8], which describes the analytical and numerical approaches towards the solution for such problems. The TE is defined as a deviation from a perfect motion of gears, which can be caused by different reasons: misalignments, teeth deflection under load, manufacturing inaccuracies and others. All these factors constitute for mechanical transmission complex dynamic behavior and should be taken into account during the design phase. One approach to solve this problem is based on the Finite Element Method (FEM), which is capable of solving accurately even the most complicated design cases. However, its major drawback, i.e., the simulation time, precludes its application to demanding problems analyzed in the time domain. In Ref. [9], the authors took the endeavor to simulate gear interaction by using comprehensive FEM approach and to tune specially prepared one degree-of-freedom (DoF) models, to be able to represent its complex nonlinear behavior. As they reported in the paper, they were able to obtain either good correlation with the results in frequency domain or satisfying meshing stiffness representation, but not both simultaneously. Similar studies were described in Ref. [10], in which the authors took the efforts aimed at correlating the results from FEM-based and analytical, one DoF dynamic models with the experiments.

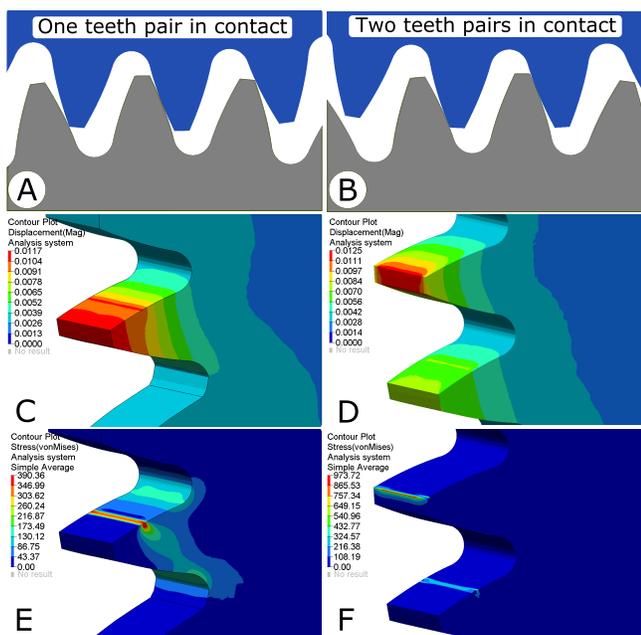


Figure 1. Analysis of gear rotation angle, for which one (A) and two (B) pairs of teeth stay in contact. Figures C and D show the teeth deflection pattern, while local Hertzian stress is presented in figures E and F.

In this paper we present a numerical study on the dynamics of a spur gear pair, which was carried out on a multibody (MB) model by implementation of a methodology reported in Ref. [11] and [12]. The utilized approach is capable of analyzing complicated geared drivelines under variable loads and operating conditions within computational time, which is orders of magnitude shorter than in the case of the adequate

FEM-based runs.

The subsequent paragraphs describe the two phases of the proposed simulation approach: firstly, static transmission error (STE) curves, describing the gear meshing stiffness, are derived by analyzing high fidelity finite element (FE) models of the gears; subsequently, time-efficient MB simulations of gear dynamics under different operating conditions are achieved by using a specifically defined user force element that allows to take the meshing stiffness variability into account thanks to the previously calculated static transmission curves. It is shown that the utilized MB technique can be successfully used for dynamic simulation of gear meshing and is able to capture its nonlinear, load-dependent behavior. The described analyses were carried out in velocity run-up conditions, for the models loaded by torque of variable values. Because of the required computational time, the same results would not be attainable using the FEM approach.

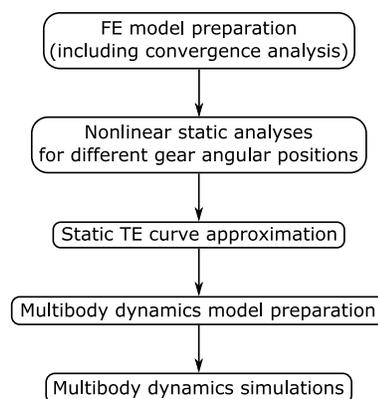


Figure 2. Flowchart of the proposed methodology for gear dynamic simulation.

The paper is structured in the following manner: Section II summarizes the proposed approach, describing in short each activity which must be taken to carry out time-efficient gearbox simulations. Section III provides a detailed description of the prepared FE model and provides a description of the mesh convergence analysis. Section IV presents and compares different methods of calculating the TE curves, allowing for choosing the most efficient approach for STE estimation. Section V describes the implementation of the described methodology in order to understand the dynamic behavior of a spur gear pair in run-up conditions.

II. GEAR DYNAMIC SIMULATION METHODOLOGY

Based on the considerations presented in Section I, we can now describe the methodology for gear dynamic simulations, which is proposed in this paper. For the sake of clarity, the flowchart of the simulation procedure is presented in Figure 2. As mentioned above, the elaborated procedure is based on derivation of a STE curve, which is computed using FEM simulations. As shown in the subsequent paragraphs, nonlinear static analyses are sufficient to describe the desired behavior of the analyzed system. This operation is executed iteratively, for a set of discrete gear angular positions. Following, the STE curve is derived by approximation of the results obtained in the preceding phase. This constitutes an input for the MB simulations, which are carried out on a MB model, prepared using the contact formulation described in Ref. [11] and [12].

MB model prepared in this manner can be used for fast and accurate dynamic simulations of complex gear trains, requiring relatively short computation time.

III. FE MODELING AND ANALYSIS OF THE SIMULATED SYSTEM

The analyses described in this paper were carried out on a pair of meshing, high-precision, identical spur gears, with involute tooth profile, for which Table 1 summarizes the main geometrical data and design specifications.

TABLE I. GEAR DATA AND DESIGN SPECIFICATIONS.

Parameter	Value
Number of teeth	57
Normal module	2.60mm
Normal pressure angle	20deg
Tip diameter	154.50mm
Root diameter	141.70mm
Facewidth	23mm
Normal circular tooth thickness at theoretical pitch circle	3.78mm
Contact Ratio	1.45

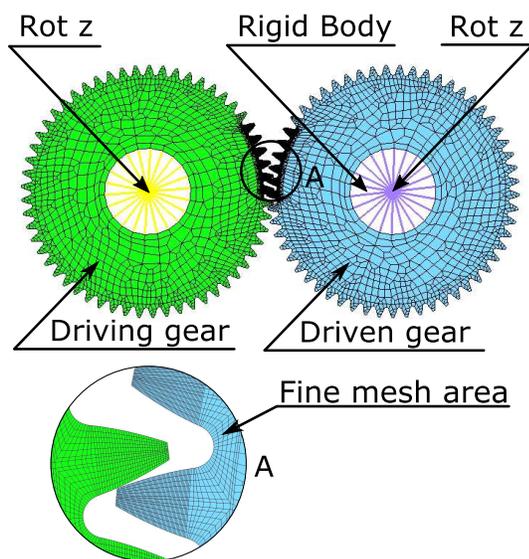


Figure 3. Three-dimensional FE model of a spur gear pair used in the described analyses.

The FEM models of a gear pair, which are depicted in Figure 3, were built using the first-order, eight-noded hexagonal elements. The deformation of this type of finite element is described by three translational DoF in each corner node, resulting in 24 DoFs per element. The shape functions describing the stress field in the elements were linear. In order to prevent from hourglass deformations (numerical, erroneous element deformations), the element stress was retrieved by full integration scheme. The model preparation and simulations were performed in Altair HyperWorks software. Two types of analyses were carried out: implicit statics with geometrical nonlinearities with Altair OptiStruct and explicit dynamic simulations with Altair Radioss. Depending on the simulation type, two distinct modeling strategies were used [13].

In the case of the explicit dynamic analyses, the contact between the interacting elements was modeled by using the interface type 24, with node to surface definition and Coulomb friction ($\mu = 0.3$). The kinematic constraints applied to the

gear centers allowed only for rotational displacement. The loading torque was generated by a viscous damper connected to the center of rotation (CoR) of the driven gear. A proper rotational velocity of the driving gear CoR was set through the defined boundary conditions.

In the case of the static analyses, geometrical nonlinearities were taken into account. This assured high accuracy of the results. In the case of these computations, the contact between the teeth surfaces was modeled by means of the interface type 7. The loading torque of an appropriate value was applied to the CoR of the driving gear, while the rotation center of the driven gear was fixed in all DoFs.

A. FE mesh convergence analysis

In order to reproduce the local phenomena of the simulated system (i.e., deflection and Hertzian stress) while keeping the computational time as low as possible, the model was divided into two areas of coarse and fine mesh. The former was used in the region where there was no interaction between the gear teeth and, hence, no local stress and contact pressure concentrations were present. In this region, the mesh size was set up to 8 mm. In the area in which the two gears interacted with each other, the FE mesh size was decreased. In order to understand the significance of the FE element size, a convergence analysis was carried out, showing the sensitivity of the STE values towards this parameter. Figure 4 depicts the chosen results of these trials: based on these findings it was decided to set the smallest element edge length to 0.09 mm. As depicted in Figure 4, the STE curve obtained for this element size was qualitatively and quantitatively equal to the equivalent calculated using the model described by a finer mesh (element edge length of 0.035mm along a tooth profile).

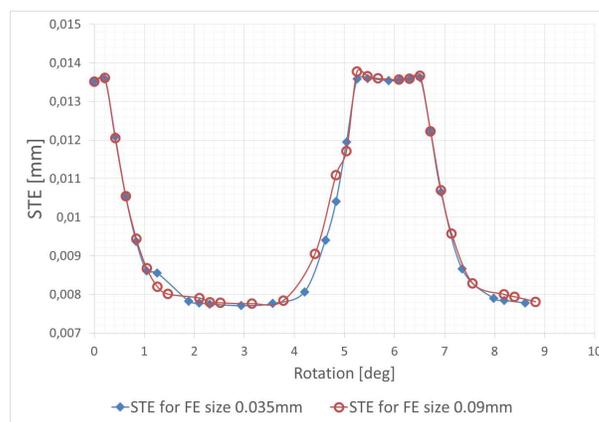


Figure 4. The results of the convergence analysis showing the STE curves obtained for different FE mesh size (torque $T=350\text{Nm}$).

For the NV assessment, the most important outcome of the convergence analysis is the peak-to-peak value of the STE curves and the contact ratio, which is the average number of meshing tooth pairs and can be derived based on the STE curve shape. These two parameters constitute for the internal excitation force of a gearbox, in its operational conditions. As shown in Figure 4, both of these characteristics were covered accurately by the tested models.

The TE curves were calculated by subtracting the angle of rotation (AoR) of the driven gear from the AoR of the driving gear, according to (1):

$$TE = (\theta_P + \theta_G)r_B \quad (1)$$

where θ_P and θ_G are the AoR for the driving and of the driven gear respectively, and r_B denotes the radius of the base circle of the gears.

IV. METHODS TO COMPUTE STE CURVES

In order to use the MB approach described in the Introduction section, an STE curve needs to be provided by the user as an input. The generation of this curve can be done by FE approach. In order to understand if the time efficient nonlinear static approach can result in an accurate estimation of the STE curve, it was compared with the detailed explicit dynamic simulation outcomes. The significant drawback of the latter is the computational time, which depends on the FE mesh size and therefore in the case of the described type of simulations it can be very long.

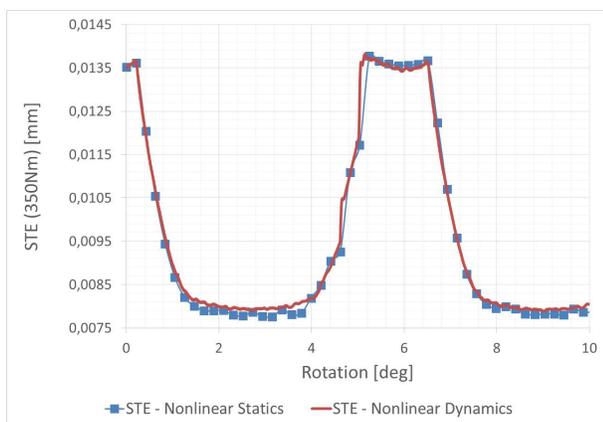


Figure 5. STE for 350Nm torque - comparison between static and dynamic approach.

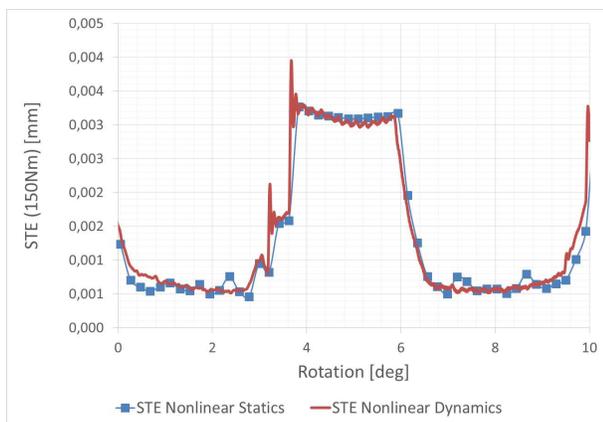


Figure 6. STE for 150Nm torque - comparison between static and dynamic approach.

In the case of the dynamic simulations, the STE curves were obtained by simulating gear rotation with a velocity run-up from zero to a very low, constant rotational speed value (10 rpm), which allowed neglecting inertia contribution to the results. The procedure involving nonlinear static simulations was based on a number of FEM implicit analyses carried out for a discrete rotation angles of the gear pair. The angle step

of the driving gear rotation was set to 0.21 deg, which resulted in 30 points per one meshing cycle. This number guaranteed a smooth representation of the STE curve, requiring a relatively low number of simulations, in the same time.

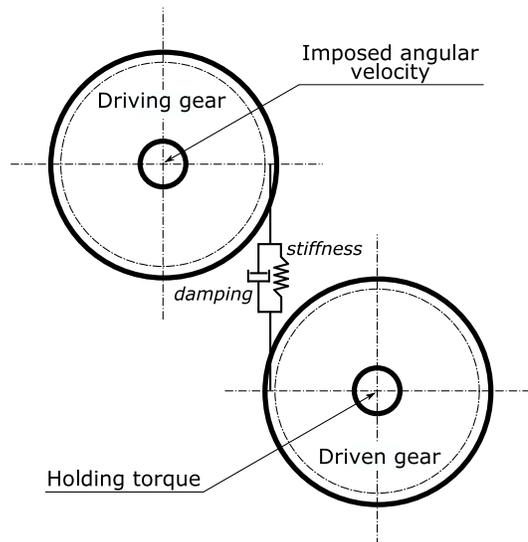


Figure 7. Diagram of a MB model of the simulated gear pair, showing the viscoelastic contact element between the gear teeth (based on [11] and [12]).

The comparison between the statically and the dynamically generated STE curves is shown in Figure 5 and Figure 6. Two different loading torque values equal to 150Nm and 350Nm were used in the simulations. It can be seen from the figures that both FEM methods resulted in quantitatively and qualitatively correlated results. The significant difference, however, lied in the computational time needed for the analysis to converge, which was equal to ca. 150 minutes and 3330 minutes for nonlinear static and dynamic analysis, respectively.

The obtained results allowed to choose the most suitable method for the generation of the STE curves, which were needed for the subsequent MB analyses, carried out with the approach proposed in Ref. [11] and [12]. Since the approach based on nonlinear static FEM calculations was significantly more time efficient, it was selected to generate the input data for the MB simulation.

V. MULTIBODY GEAR ANALYSIS

The implementation of the multibody technique requires that the gear geometrical description (i.e., number of teeth, diameter, pressure angle) is provided simultaneously with the data describing STE, spatial misalignments, torque variability and microgeometry modifications. All of this information influences the dynamic response of the system, allowing computing the dynamic transmission error (DTE). The presented in Figure 7 and Figure 8 MB model was built in the Siemens LMS Virtual.Lab 13.1 environment [14], in a manner which was consistent with the FEM representation of the gear pair described above. The constraints used in the MB model allowed rotations of the gear centers of rotation. The loading torque was applied to the driven gear, while the rotational velocity was imposed for the driving gear. Moreover, as shown in Figure 7, the implemented MB gear contact element assumed viscoelastic behavior of interacting teeth.

The DTE curves presented in Figure 9 and Figure 10 were obtained under rotational speed linear run-up conditions, from 0 rpm to 3500 rpm and for two values of the loading torque: 150Nm and 350Nm. The simulated operating time was set to 70 seconds. Since the two gears had a number of teeth equal to 57, the type of excitation due to meshing stiffness variation along one meshing cycle could be considered as a multi-harmonic frequency sweep. With the maximum rotational velocity, it generated an excitation fundamental frequency of 3325 Hz.

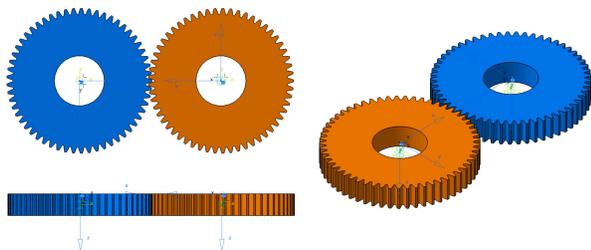


Figure 8. Gears model in multibody dynamics simulation environment.

By looking at the envelopes of the obtained DTE curves, some local amplitude amplifications can be seen. These corresponded with the gear pair resonance, excited by the TE, due to gear meshing stiffness variability. In the case of the system loaded by the 150Nm torque (Figure 9), the most significant DTE amplifications were found at 116Hz, 438Hz, 597Hz, 907Hz and 1804Hz, while for the same mechanical transmission loaded by the 350Nm torque (Figure 10), the DTE amplifications were found at 122Hz, 479Hz, 650Hz, 980Hz and 1958Hz. The differences between these values are summarized in Table II.

Indeed, in the utilized modeling method, the gear meshing stiffness variation is derived from the STE curve supported by the user and shows a stiffening behavior which is typical of contact problems [15]. Based on its periodic, time-varying characteristics it was possible to cover the nonlinearities present in the described system. In the discussed analyses, the STE curves depicted in Figure 5 and Figure 6 were used.

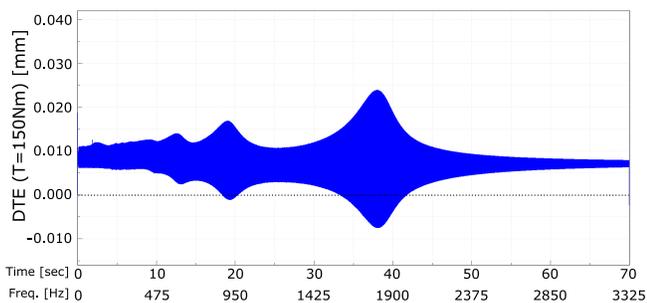


Figure 9. DTE calculated in run-up conditions, for spur gears pair loaded by 150Nm torque.

The obtained results prove the capabilities of the used three-dimensional MB modeling methodology, based on the FE gear contact formulation described in Section III. The depicted in Figure 9 and Figure 10 gear pair behavior could not be captured by consideration of a standard one-dimensional or planar, linear representation of interacting teeth, described in the introductory paragraph - Section I. This is because

TABLE II. THE DTE AMPLIFICATIONS IDENTIFIED FOR THE ANALYSED SPUR GEAR PAIR IN RUN-UP CONDITIONS.

DTE amplification id	Loading torque		Difference	
	150Nm	350Nm	Δ	Δ [%]
1	116	122	6	5.17
2	438	479	41	9.36
3	597	650	54	9.05
4	907	980	72	7.94
5	1804	1958	153	8.48

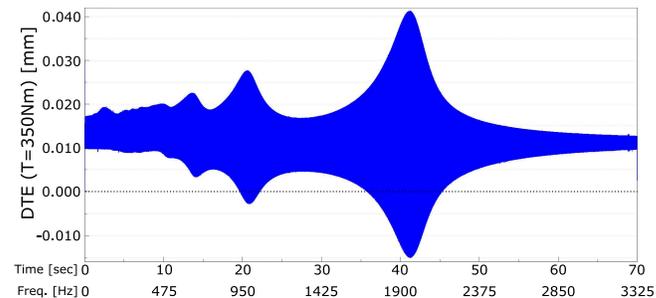


Figure 10. DTE calculated in run-up conditions, for spur gears pair loaded by 350Nm torque.

these simplified models neglect the contact stiffness variability along the tooth profile. On the other hand, if applied, more comprehensive FEM analyses could have resulted in the same results, as it was shown in Ref. [2]. However, in the presented case, the computational time was significantly lower compared with a possible dynamic FEM calculations (e.g., shown in Section III), and was equal to ca. 36 minutes on a standard desktop computer: i7@3.2GHz CPU, 32GB RAM.

VI. CONCLUSION AND FUTURE WORK

It was shown in the paper that it is possible to analyze the nonlinear behavior of a meshing gear pair, using a simplified, yet efficient MB analysis technique. By application of this method, the time needed for a comprehensive analysis of a driveline system is orders of magnitude lower, when compared with the equivalent dynamic FEM calculations. This allows to support the industrial design process with an accurate and time-efficient simulation tool, which would make feasible the assessment of different gear design variants during the virtual prototyping phase. Moreover, the proposed methodology for the DTE estimation can be effectively employed for the purposes of driveline structural optimization, which requires a number of iteratively carried out simulations and hence, can be very time consuming. Because of the numerical complexity, this would not be possible using the classical FEM-based approach. The activity scheduled as the nearest future step of the research work presented in this paper is aimed at improving gearbox noise and vibration characteristics, by application of teeth profile microgeometry modifications. By implementation of the described methodology and because of its time-efficiency, the optimization approach can be more comprehensive, including different load cases and operational conditions. This will ease convergence to a global optimum of the imposed problem.

ACKNOWLEDGMENT

The research activities described in this paper were funded within the People Programme (Marie Curie Actions) of the

7th Framework Programme of the European Union FP7/2007-2013, under the contract with the Research Executive Agency (REA) n. 324336 and the related research project DEMETRA Design of Mechanical Transmissions: Efficiency, Noise and Durability Optimization (www.fp7demetra.eu).

REFERENCES

- [1] S. Shweiki, A. Palermo, A. Toso, D. Mundo, and W. Desmet, "Effects of center distance and microgeometry on the dynamic behaviour of a spur gear pair," in Proceedings of the International Conference on Engineering Vibration (ICOEV 2015) September 7–10, 2015, Ljubljana, Slovenia, 2015, pp. 119–130.
- [2] F. L. Litvin and F. A., Eds., Gear geometry and applied theory. Cambridge University Press, 2004, ISBN: 0-521-81517-7.
- [3] H. N. Özgüven and D. R. Houser, "Dynamic analysis of high speed gears by using loaded static transmission error," Journal of Sound and Vibration, vol. 125.1, 1988, pp. 71–83, ISSN: 0022-460X.
- [4] G. Liu and R. G. Parker, "Dynamic modeling and analysis of tooth profile modification for multimesh gear vibration," Journal of Mechanical Design, vol. 130, no. 12, 2008, p. 121402.
- [5] D. R. Houser, J. Harianto, and Y. Ueda, "Determining the source of gear whine noise," Gear Solutions, vol. 1623, 2004, pp. 17 – 22.
- [6] J. D. Smith, Gear noise and vibration. CRC Press, 2003.
- [7] S. Theodossiades and S. Natsiavas, "Non-linear dynamics of gear-pair systems with periodic stiffness and backlash," Journal of Sound and vibration, vol. 229, no. 2, 2000, pp. 287–310.
- [8] Q. Ma and A. Kahraman, "Period-one motions of a mechanical oscillator with periodically time-varying, piecewise-nonlinear stiffness," Journal of Sound and Vibration, vol. 284, 2005, pp. 893–914.
- [9] R. G. Parker, S. M. Vijayakar, and T. Imajo, "Non-linear dynamic response of a spur gear pair: modelling and experimental comparisons," Journal of sound and vibration, vol. 237, no. 3, 2000, pp. 435–455.
- [10] V. K. Tamminana, A. Kahraman, and S. Vijayakar, "A study of the relationship between the dynamic factors and the dynamic transmission error of spur gear pairs," Journal of Mechanical Design, vol. 129, 2007, pp. 75–84.
- [11] A. Palermo, D. Mundo, R. Hadjit, and W. Desmet, "Multibody element for spur and helical gear meshing based on detailed three-dimensional contact calculations," Mechanism and Machine Theory, vol. 62, 2013, pp. 13–30.
- [12] A. Palermo, D. Mundo, R. Hadjit, and Desmet, "Effects of variable loads and misalignments on gear noise and vibration through multibody simulation based on transmission error," in Proceedings of the International Conference on Noise and Vibration Engineering (ISMA2012) and 4th International Conference on Uncertainty in Structural Dynamics (USD2012) September 17–19, 2012, Leuven, Belgium, 2012, pp. 3995–4006.
- [13] R. Version, "5.1, radioss theory manual, altair engineering," Inc., Troy, MI, 2008.
- [14] Siemens PLM Software, LMS Virtual.Lab 13.2, <http://www.siemens.com/plm/lms>, 2015.
- [15] K. L. Johnson and K. L. Johnson, Contact mechanics. Cambridge university press, 1987.

Alternative System Design Based on Reliability Analysis and Simulation

Merve Uzuner, Berna Dengiz
 Industrial Engineering Department
 Baskent University, Ankara, Turkey
 e-mails: {muzuner, bdengiz}@baskent.edu.tr

Abstract—The purpose of this study is the integration of reliability-availability analysis and simulation modeling to improve the productivity of the battery production system. The throughput of the system is increased by using integrated reliability-availability and simulation approach together and the proposed system design is more reliable than the current design. Moreover, system availability provides a valuable input for the company to realize the production plan for the future.

Keywords-simulation modeling; reliability-availability analysis; battery production system.

I. INTRODUCTION

This paper presents a real case study of a multi-stage production line in battery production factory. In today's highly competitive industry, a company must be able to adapt to its customers' ever changing needs and improve the reliability of their system and products in order to survive. The battery production system of a company in Ankara, Turkey, is a complex system as is often the case in manufacturing environments, and so, to predict the throughput and to investigate the stochastic behavior of this system, simulation model could not be used alone. Although most operation units have stochastic operation times, curing unit has much longer constant operation time than the other operation units. Therefore, under its dominant effect, the simulation model works like a deterministic model not a stochastic. Although failure time, mean time to failure and down time are known for each machine working at all stations failures could not be considered in the simulation model. Because simulation run time of this very complex system already takes too long time such as 60 hrs even without considering failures. Therefore, simulation model of this system is built without failures. However, it is needed an additional method to imitate the stochastic nature of this system to be able to get true response from the system to improve system performance.

II. METHODOLOGY

In today's automated systems, to analyze production rates and system qualifications, reliability is one of the most important metrics of components and systems. Reliability could be analyzed in two main headings, product reliability and system reliability. Product reliability is the probability that a product will operate properly for a specified period of time. System reliability can be computed based on system configuration where all components are connected in series, in parallel or as networks. Some metrics related to reliability of components and systems are mean time to failure [1], the failure rates of components or system [2], the availability of components or system [3], and the repair rate of system [4].

Companies should present more reliable products in current competitive market; otherwise, customer satisfaction could not be provided and products are not preferred. Therefore, reliability and availability analysis in production fields have attracted interest recently, although in the literature, reliability analysis and availability analysis generally have been considered in electronic fields.

The battery is the main power component of the vehicles. Any failure of the battery (not working appropriately) could cause problems about vehicle or driver. It is desired that it has high reliability and availability. Therefore, reliability computation of production systems is important.

In this study, we propose to use the integration of reliability-availability analysis and simulation modeling to improve the productivity of the system because of system characteristics. In addition, overall system availability will provide a valuable input for realization of production plan for the future. Hence, in practice, production planning department can find valuable evaluation value to improve the system productivity. To the best of our knowledge the proposed method in this work is the first application in this area.

This study will proceed as follows: Firstly, a new system design is obtained based on reliability and availability analysis of the current system. The reliability block diagram method is used for the computation of reliability of existing and proposed complex battery production system. Secondly, we investigate the system behavior to find out bottlenecks and see the throughput rates under assumption of no failures with the simulation model.

III. CASE STUDY

In the eleven stations, only machines are taken into account for system reliability and availability analysis. In each operation units, all components are defined and their failure times and repair times are obtained from historical data. Failure distributions and their parameters are estimated and repair rate of the system are calculated [5]. The battery production system is analyzed and reliability block diagram is drawn considering their series or parallel configuration. The reliability block diagram of current battery production system is shown in Figure 1.

System reliability and system availability is computed [6]. The reliability equation of current system is obtained based on [7]. The system availability of current system is 0,890309 per month.

System reliability and availability is calculated again for new proposed system. System availability for alternative system is 0,899266 per month.

Using the reliability and availability analysis, new system

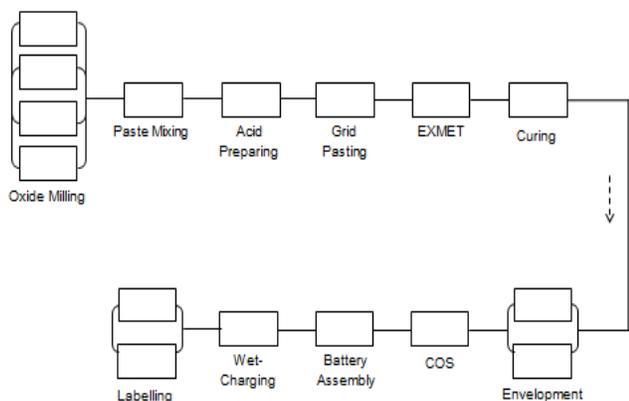


Figure1. Reliability Block Diagram of current battery production system.

is developed to improve the system productivity. To get a more reliable alternative system some stations (components) are added to the current system. Thus, a new series and parallel alternative system is designed.

Secondly, we investigate the system behavior to find out bottlenecks and see the throughput rates under assumption of no failures with the simulation model of existing and proposed alternative systems. In our approach, a simulation model is constructed using discrete event simulation. The ARENA simulation package [8] is preferred for its capability. Actual data from the production was utilized to determine the inputs for each process. Simulation model is modified and used to estimate the throughput of this alternative system. It is 7411 batteries per month.

Finally, using the simulation result and system availability value, expected production rates are obtained considering failures as follows. It is calculated that 6384 batteries can be produced each month by the current production system. The actual throughput considering failures is 6664,46 in the proposed new system. It can be said

that using integrated reliability-availability and simulation approach together it can be obtained more reliable alternative system design with 6664 throughput (battery), which is more than the current system.

IV. CONCLUSION

This paper presented a new method to improve the productivity of the battery production system of a company in Ankara, Turkey. Because of system characteristics, reliability-availability analysis and simulation approach are used together to analyze system and obtain throughput (battery). Finally, it is seen that the proposed new system design is more reliable than the existing system design and it is produced more battery than the existing system.

REFERENCES

- [1] M. Rausand and A. Hoyland, System Reliability Theory, Models, Statistical Methods, And Applications, 2nd ed., Wiley Interscience, p.8, 2004.
- [2] E. A. Elsayed, Reliability Engineering, Addison Wesley Longman, p.6, 1996.
- [3] E. A. Elsayed, Reliability Engineering, Addison Wesley Longman, p.180, 1996.
- [4] M. Rausand and A. Hoyland, System Reliability Theory, Models, Statistical Methods, And Applications, 2nd ed., Wiley Interscience, p.189, 2004.
- [5] Johnson Space Center, Mean time to repair predictions. Available from: oce.jpl.nasa.gov/practices/at2.pdf [Oct. 2015].
- [6] M. Rausand and A. Hoyland, System Reliability Theory, Models, Statistical Methods, And Applications, 2nd ed., Wiley Interscience, p.7, 2004.
- [7] V. Faraci, "Calculating Failure Rates Series/Parallel Networks", The Journal of Alion's, System Reliability Center, 2006.
- [8] A. M. Law, Simulation Modeling&Analysis, 4th ed., McGraw-Hill, p.70, 2007.