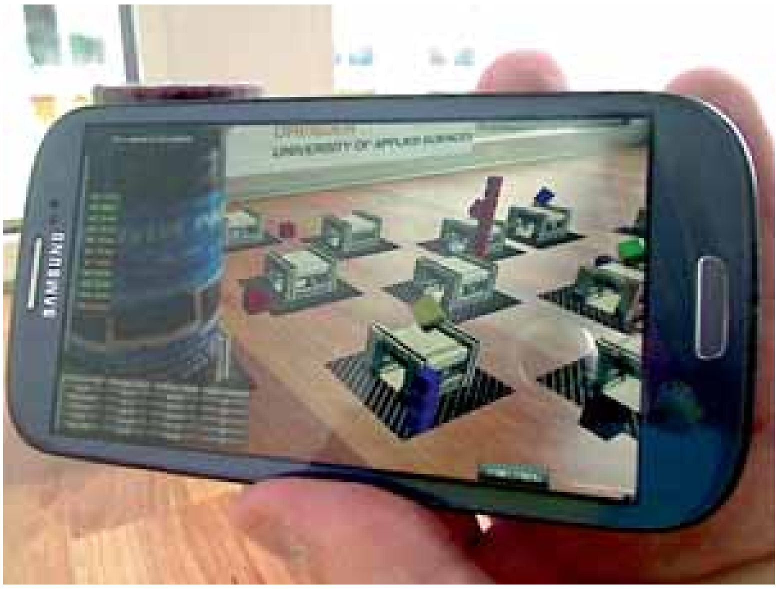
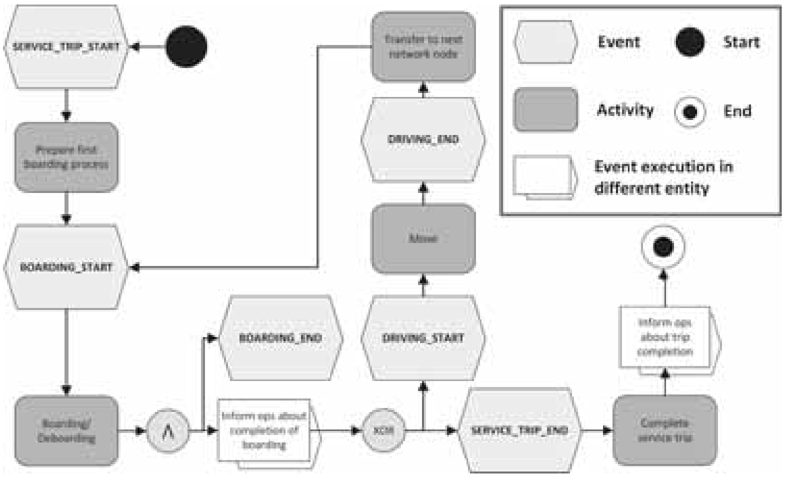
SNE SIMULATION NOTES EUROPE




SERVICE_TRIP_START
Prepare first boarding process
BOARDING_START
Boarding/ Deboarding
∧
BOARDING_END
Inform ops about completion of boarding
XOR
Transfer to next network node
DRIVING_END
Move
DRIVING_START
SERVICE_TRIP_END
Complete service trip
Inform ops about trip completion
Event
Start
Activity
End
Event execution in different entity

Journal on Developments and Trends in Modelling and Simulation

EUROSIM Scientific Membership Journal

Vol. 27 No.1, March 2017 ISSN Online 2306-0271 DOI 10.11128/sne.27.1.1036

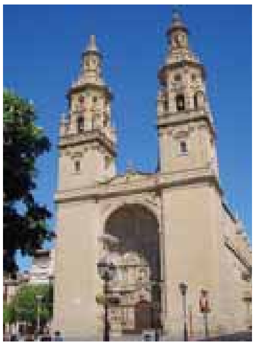
TU Verlag
an der Technischen Universität Wien

ARGESIM

EUROSIM 2019

9th EUROSIM Congress on Modelling and Simulation

La Rioja, Logroño, Spain, July 2019



EUROSIM Congresses are the most important modelling and simulation events in Europe. For EUROSIM 2019, we are soliciting original submissions describing novel research and developments in the following (and related) areas of interest: Continuous, discrete (event) and hybrid modelling, simulation, identification and optimization approaches. Two basic contribution motivations are expected: M&S Methods and Technologies and M&S Applications. Contributions from both technical and non-technical areas are welcome.

Congress Topics The EUROSIM 2019 Congress will include invited talks, parallel, special and poster sessions, exhibition and versatile technical and social tours. The Congress topics of interest include, but are not limited to:

Intelligent Systems and Applications
Hybrid and Soft Computing
Data & Semantic Mining
Neural Networks, Fuzzy Systems & Evolutionary Computation
Image, Speech & Signal Processing
Systems Intelligence and Intelligence Systems
Autonomous Systems
Energy and Power Systems
Mining and Metal Industry
Forest Industry
Buildings and Construction
Communication Systems
Circuits, Sensors and Devices
Security Modelling and Simulation

Bioinformatics, Medicine, Pharmacy and Bioengineering
Water and Wastewater Treatment, Sludge Management and Biogas Production
Condition monitoring, Mechatronics and maintenance
Automotive applications
e-Science and e-Systems
Industry, Business, Management, Human Factors and Social Issues
Virtual Reality, Visualization, Computer Art and Games
Internet Modelling, Semantic Web and Ontologies
Computational Finance & Economics

Simulation Methodologies and Tools
Parallel and Distributed Architectures and Systems
Operations Research
Discrete Event Systems
Manufacturing and Workflows
Adaptive Dynamic Programming and Reinforcement Learning
Mobile/Ad hoc wireless networks, mobicast, sensor placement, target tracking
Control of Intelligent Systems
Robotics, Cybernetics, Control Engineering, & Manufacturing
Transport, Logistics, Harbour, Shipping and Marine Simulation

Congress Venue / Social Events The Congress will be held in the City of Logroño, Capital of La Rioja, Northern Spain. The main venue and the exhibition site is the University of La Rioja (UR), located on a modern campus in Logroño, capital of La Rioja, where 7500 students are registered. The UR is the only University in this small, quiet region in Northern Spain. La Rioja is where the Monasteries of San Millán de la Cogolla, cradle of the first words written in the Spanish language, are situated, sites included in UNESCO's World Heritage List in 1996. Of course, social events will reflect this heritage – and the famous wines in la Rioja.

CongressTeam: The Congress is organised by CAE CAE-SMSG, the Spanish simulation society, and Universidad de la Rioja.

Info: Emilio Jiménez, EUROSIM President, emilio.jimenez@unirioja.es
Juan Ignacio Latorre, juanignacio.latorre@unavarra.es

www.eurosim.info

Editorial

Dear Readers – *SNE Volume 27 (2017) provides novelties in content and organisation. SNE has updated and sharpened aims and scope. First, SNE puts emphasis on the whole simulation process - from modelling to experiment analysis, from implementation to verification, from validation to identification, from numerics to visualization, and therefore SNE asks authors to present their contribution within this context. Of course, authors may concentrate on modelling in a certain application area, but they should mention the embedding into the full simulation process (simulation circle). Second, SNE follows the recent developments and trends of modelling and simulation in new and/or joining application areas, as complex systems and big data. Third, SNE also invites authors to submit contributions on comparative investigations, as comparison of modelling approaches, effectiveness of different implementations, benchmarks and comparisons in methodology and application, etc. Here, SNE plans to become a forum for comparative studies and benchmarks, with data space for definition, documentation, etc. And fourth, SNE welcomes also contributions on education in/for/with simulation, also in context with contributions on comparative studies in modelling and simulation methods. Organisationally, SNE will set up a new web server – ready midst of 2017, and SNE has found an appropriate solution for Print SNE. We are happy, that TU Verlag (TU Wien Publisher - www.tuverlag.at) will provide Print-on-Demand for SNE issues, whereby each issue is identified by an ISBN number from TU Verlag – starting with this issue SNE 27(1).*

I would like to thank all authors for their contributions to SNE 27(1) showing the broad variety of simulation in context with the updated aims and scope, from augmented realty via basic discrete simulation and parallel smart-grid simulation to simulation in education. And thanks to the editorial board members for review and support, and to the organizers of the EUROSIM conferences for co-operation in post-conference contributions. And last but not least thanks to the SNE Editorial Office for layout, typesetting, preparations for printing, with special thanks for support in relaunch of SNE webserver.

Felix Breitenecker, SNE Editor-in-Chief, eic@sne-journal.org; felix.breitenecker@tuwien.ac.at

Contents SNE 27(1)

Online SNE 27(1) DOI 10.11128/sne.27.1.1036
Print SNE 27(1) Print-on-Demand **ISBN 978-3-903024-50-2**

SNE Contact & Info

SNE Online ISSN 2306-0271, SNE Print ISSN 2305-9974
→ *www.sne-journal.org*
✉ *office@sne-journal.org, eic@sne-journal.org*

✉ SNE Editorial Office, Andreas Körner
ARGESIM/Math. Modelling & Simulation Group,
Vienna Univ. of Technology /101,
Wiedner Haupstrasse 8-10, 1040 Vienna , Austria

SNE Simulation Notes Europe

WEB: → *www.sne-journal.org*, DOI prefix 10.11128/sne

Scope: Developments and trends in modelling and simulation in various areas and in application and theory; comparative studies and benchmarks (documentation of ARGESIM Benchmarks on modelling approaches and simulation implementations); modelling and simulation in and for education, simulation-based e-learning; society information and membership information for EUROSIM members (Federation of European Simulation Societies and Groups).

Editor-in-Chief: Felix Breitenecker, Vienna Univ. of Technology, Math. Modelling and Simulation Group
✉ *Felix.Breitenecker@tuwien.ac.at*, ✉ *eic@sne-journal.org*

Layout / Administration: A. Körner, A. Mathe, J. Tanzler, R. Leskovar, et al.; ✉ *office@sne-journal.org*

Print SNE and Print-on-Demand: Grafisches Zentrum and TU-Verlag, TU Wien, Wiedner Hauptstrasse 8-10, 1040, Vienna, Austria – *www.tuverlag.at*

Publisher: ARGESIM Arbeitsgemeinschaft Simulation News
c/o Math. Modelling and Simulation Group,
Vienna Univ. of Technology / 101, Wiedner Hauptstrasse 8-10,
1040 Vienna, Austria; *www.argesim.org*, ✉ *info@argesim.org*
on behalf of ASIM *www.asim-gi.org* and EUROSIM
→ *www.eurosim.info*

© ARGESIM / EUROSIM / ASIM 2017

SNE - Aims and Scope

Simulation Notes Europe (SNE) provides an international, high-quality forum for presentation of new ideas and approaches in simulation - from modelling to experiment analysis, from implementation to verification, from validation to identification, from numerics to visualisation - in context of the simulation process.

SNE seeks to serve scientists, researchers, developers and users of the simulation process across a variety of theoretical and applied fields in pursuit of novel ideas in simulation and to enable the exchange of experience and knowledge through descriptions of specific applications. **SNE** follows the recent developments and trends of modelling and simulation in new and/or joining application areas, as complex systems and big data. **SNE** puts special emphasis on the overall view in simulation, and on comparative investigations, as benchmarks and comparisons in methodology and application. For this purpose, **SNE** documents the **ARGESIM Benchmarks** on *Modelling Approaches and Simulation Implementations* with publication of definitions, solutions and discussions. SNE welcomes also contributions in education in/for/with simulation.

A News Section in **SNE** provides information for **EUROSIM** Simulation Societies and Simulation Groups.

SNE, primarily an electronic journal, follows an open access strategy, with free download in basic layout. **SNE** is the official membership journal of **EUROSIM**, the *Federation of European Simulation Societies and Simulation Groups* – www.eurosim.info. Members of **EUROSIM** societies are entitled to download **SNE** in an elaborate and extended layout, and to access additional sources of benchmark publications, model sources, etc. **Print SNE** is available for specific groups of **EUROSIM** societies, and starting with Volume 27 (2017) as print-on-demand from TU Verlag, TU Wien. **SNE** is DOI indexed by CrossRef, identified by DOI prefix 10.11128, assigned to the **SNE** publisher **ARGESIM** ().

Author's Info. Individual submissions of scientific papers are welcome, as well as post-conference publications of contributions from conferences of **EUROSIM** societies. **SNE** welcomes special issues, either dedicated to special areas and/or new developments, or on occasion of events as conferences and workshops with special emphasis.

Authors are invited to submit contributions which have not been published and have not being considered for publication elsewhere to the **SNE** Editorial Office.

SNE distinguishes different types of contributions (*Notes*), i.e.

- **TN** Technical Note, 6 – 10 p.
- **PN** Project Note 6 – 8 p.
- **SW** Software Note , 4 – 6 p.
- **EN** Education Note –6 – 8 p.
- **SN** Short Note, max. 6 p.
- **BN** Benchmark Note, 4 – 8 p.
- **ON** Overview Note *ON* – only upon invitation, up to 14 p.

Further info and templates (doc, tex) at **SNE's** website.
www.sne-journal.org

SNE Editorial Board

SNE - Simulation Notes Europe is advised and supervised by an international scientific editorial board. This board is taking care on peer reviewing of submission to SNE. At present, the board is increasing (see website www.sne-journal.org):

David Al-Dabass, *david.al-dabass@ntu.ac.uk,*
Nottingham Trent University, UK

Felix Breitenecker, *Felix.Breitenecker@tuwien.ac.at*
Vienna Univ. of Technology, Austria, Editor-in-chief

Maja Atanasijevic-Kunc, *maja.atanasijevic@fe.uni-lj.si*
Univ. of Ljubljana, Lab. Modelling & Control, Slovenia

Aleš Belič, ales.belic@sandoz.com
Sandoz / National Inst. f. Chemistry, Slovenia

Peter Breedveld, *P.C.Breedveld@el.utwente.nl*
University of Twenty, Netherlands

Agostino Bruzzone, *agostino@itim.unige.it*
Universita degli Studi di Genova, Italy

Francois Cellier, *fcellier@inf.ethz.ch*
ETH Zurich, Switzerland

Vlatko Čerić, *vceric@efzg.hr*
Univ. Zagreb, Croatia

Russell Cheng, *rchc@maths.soton.ac.uk*
University of Southampton, UK

Roberto Cianci, *cianci@dime.unige.it*, Math. Eng. and Simulation,
Univ. Genova, Italy

Eric Dahlquist, *erik.dahlquist@mdh.se*, Mälardalen Univ., Sweden

Horst Ecker, *Horst.Ecker@tuwien.ac.at*
Vienna Univ. of Technology, Inst. f. Mechanics, Austria

Vadim Engelson, vadim.engelson@mathcore.com
MathCore Engineering, Linköping, Sweden

Edmond Hajrizi, *ehajrizi@ubt-uni.net*
University for Business and Technology, Pristina, Kosovo

András Jávor, *javor@eik.bme.hu,*
Budapest Univ. of Technology and Economics, Hungary

Esko Juuso, *esko.juuso@oulu.fi*
Univ. Oulu, Dept. Process/Environmental Eng., Finland

Kaj Juslin, kaj.juslin@vtt.fi
VTT Technical Research Centre of Finland, Finland

Andreas Körner, andreas.koerner@tuwien.ac.at
Technical Univ. Vienna, E-Learning Dpt., Vienna, Austria

Francesco Longo, *f.longo@unical.it*
Univ. of Calabria, Mechanical Department, Italy

Yuri Merkuryev, *merkur@itl.rtu.lv,* Riga Technical Univ.

David Murray-Smith, *d.murray-smith@elec.gla.ac.uk*
University of Glasgow, Fac. Electrical Engineering, UK

Gasper Music, *gasper.music@fe.uni-lj.si*
Univ. of Ljubljana, Fac. Electrical Engineering, Slovenia

Thorsten Pawletta, *pawel@mb.hs-wismar.de*
Univ. Wismar, Dept. Comp. Engineering, Wismar, Germany

Niki Popper, *niki.popper@dwh.at*
dwh Simulation Services, Vienna, Austria

Kozeta Sevrani, *kozeta.sevrani@unitir.edu.al*
Univ. Tirana, Inst.f. Statistics, Albania

Thomas Schriber, *schriber@umich.edu*
University of Michigan, Business School, USA

Yuri Senichenkov, *sneyb@dcn.infos.ru*
St. Petersburg Technical University, Russia

Oliver Ullrich, oullrich@cs.1u.edu
Florida International University, USA

Siegfried Wassertheurer, *Siegfried.Wassertheurer@ait.ac.at*
AIT Austrian Inst. of Technology, Vienna, Austria

Sigrid Wenzel, *S.Wenzel@uni-kassel.de*
Univ. Kassel, Inst. f. Production Technique, Germany

A Model Factory in Augmented Reality as an Eye-Catcher at Exhibitions and Fairs

Torsten Munkelt[1*], Steven Behne[1], Markus Wacker[1], Sven Völker[2]

[1]Dresden University of Applied Sciences, Friedrich-List-Platz 1, 01069 Dresden, Germany;
*Torsten.Munkelt@htw-dresden.de

[2]Ulm University of Applied Sciences, Prittwitzstraße 10, 89075 Ulm, Germany

SNE 27(1), 2017, 1 - 8, DOI: 10.11128/sne.27.tn.10361
Received: February 5, 2017 (Selected ASIM STS 2016 Postconf. Publ.), Accepted: February 20, 2017
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Our universities required an eye-catcher at exhibitions, conferences, open houses etc. We decided in favor of a model factory in augmented reality demonstrating a queueing system, material flow, and aspects of pro-duction planning and control. When visitors look at an empty table ‘through’ their personal mobile devices, they can watch a factory producing on the table. The visitors can alter control variables and watch the factory change its behavior accordingly. We explain the underlying simulation model and its configuration, the composition and functionality of the mobile app for augmented reality, and the communication via database. Furthermore, we present a solution for the automatic generation of the model factory's layout, and we explicate how to eliminate the blurring of the model factory in augmented reality. Finally, we share experiences and user feedback from first exhibitions of the model factory and outline our plans for its future de-velopment.

Introduction

At industrial trade fairs, recruitment events and similar occasions, exhibitors compete for visitors’ attention. Most exhibitors try to contact as many visitors as possible because the overall number of contacts correlates with the number of leads. To initiate contact, an eye-catcher is needed which attracts visitors to the booth. Posters, on-screen presentations and even video clips do not serve this purpose well, since these types of media are omnipresent at fairs. The ideal eye-catcher should be physical, moving, and sending optical and acoustic signals (e.g. it should blink and beep), thus, appealing to multiple senses. According to experience, the eye-catcher often does not relate to the products or services of the exhibitor and attracts visitors nonetheless.

The Faculty of Computer Science/Mathematics of Dresden University of Applied Sciences inspired this work. The Faculty required an eye-catcher at trade fairs, exhibitions, conferences, open houses etc. The ultimate goal is to acquire new students for our universities, faculties, and courses of study.

The original idea was to build a transportable model factory. Since many universities already possess model factories illustrating industrial engineering, the new model factory was to focus on queueing systems and material flow instead. Hence, our eye-catcher illustrates some principles of production planning and control which relates the eye-catcher to our services and consequently makes it even more appealing to visitors.

Furthermore, it is required that the model factory

- can be influenced by setting structural and behavioral parameters, e.g. the number of machines, the routings, and the control strategy,
- produces perpetually, e.g. does not run empty,
- blinks and beeps occasionally,
- is easy to transport and to set up, and
- fits into the trunk of a station wagon.

When developing a concept for the model factory, the focus shifted from a physical model to a model factory in augmented reality (AR) due to several reasons: State-of-the-art technology of mobile devices and augmented reality frameworks allow the creation of an AR model factory with reasonable effort. Such a model factory is configurable according to the specific exhibition, and visitors can manipulate it directly without the risk of physical damage. Furthermore, there is currently no comparable AR model factory known, which gives us a ‘unique selling point’.

A realization of an AR model factory must fulfill the following basic requirements:

- Visitors can see the factory (machines and production orders in process) through their personal mobile devices (smartphones or tablet computers) but not with bare eyes.
- Mobile devices project the factory on the spatially restricted top of a conventional table.
- Production orders are visualized during their entire lifecycle, e.g. while they are moving from station to station, waiting in queues, and being processed at the machines.
- Visitors may set values of control parameters online on their mobile devices and observe the resulting effect on key performance indicators like adherence to delivery dates.

1 Types and Applications of Model Factories

Regarding their technical basis, there are different types of model factories: virtual factories, downscaled physical factories, and life-size physical factories. Model factories serve different purposes, e.g. evaluation, demonstration, and education.

Virtual factories are digital simulation models that do not contain any physical components. They are widely used for evaluating alternative production concepts (e.g. [1]) as well as for educational case studies (e.g. [2]), since they can represent arbitrary complex production systems and are easy to modify.

Downscaled physical model factories consist of small physical models of factories' resources. The resource models are either simple blocks or functional resources that handle simplified 'products' like small bricks. Components for creating this type of model factories are commercially available (e.g. [3]). Downscaled physical model factories typically serve educational purposes.

Life-size model factories are equipped with machines also used in real factories. Although they are usually restricted to a few workstations, these model factories require considerable space and are quite expensive. Their purpose is to demonstrate state-of-the-art technology (e.g. [4]), professional training (e.g. [5]), and research (e.g. [6]). Some of the life-size model factories use AR applications to provide manufacturing information to workers.

An investigation concerning AR based model factories did not come up with any existing solution similar to the one described in this paper.

2 Overview of the System Architecture of the AR Model Factory

Figure 1 shows an overview of the system architecture of the AR model factory.

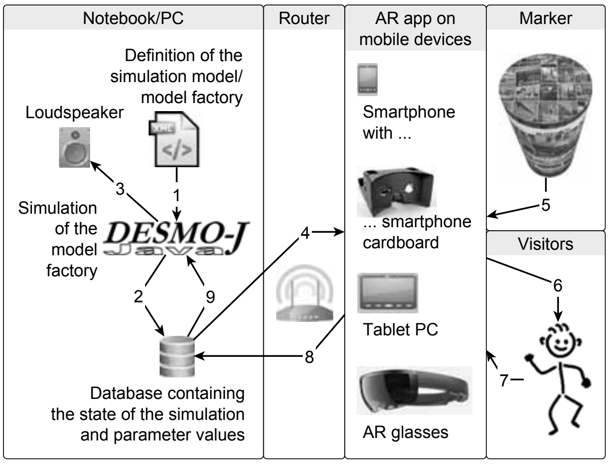

Notebook/PC
Router
AR app on mobile devices
Marker
Definition of the simulation model/ model factory
Loudspeaker
Smartphone with ...
3
1
5
Simulation of the model factory
DESMO-J
4
... smartphone cardboard
Visitors
6
2
9
Tablet PC
8
7
Database containing the state of the simulation and parameter values
AR glasses

Figure 1: Overview of the system architecture.

The system operates as follows: The simulation environment, DESMO-J, loads an XML configuration file (1), which contains all relevant parameters of the factory, e.g. machines, routings, and the product mix. DESMO-J automatically configures an executable simulation model based on these parameters. Then the simulation starts. Initially, the simulation writes master data to the database. As the simulation progresses, DESMO-J writes the current state of the model factory to the database and updates it at every change in state (2).

Acoustic signals mark certain events during the simulation (3). A WLAN router connects the mobile device to the notebook/PC. The mobile device periodically polls the current state of the model factory from the database (4). The visitor points the camera of the mobile device to the marker standing on top of the table. Now, the app on the mobile device receives the camera's picture (5) and recognizes the marker's position in relation to the position of the camera. The app overlays the picture the camera is receiving with the visualization of the model factory according to the position of the marker in relation to the position of the camera.

If the visitor looks at the table ‘through’ the mobile device, she or he will see the model factory in its current state on top of the table (6). Figure 2 shows the visitor’s view of the AR model factory.

The app updates the view in short intervals resulting in an animation of the production process. The visitor may change control parameters of the model factory via the mobile device (7). The app writes the new parameter values into the database (8), the simulation system reads the new parameter values (9) and adopts its behavior accordingly. Then, the process starts anew, and the visitor watches the changed behavior of the AR model factory on the mobile device. Several visitors may view the model factory at the same time using their personal mobile devices.

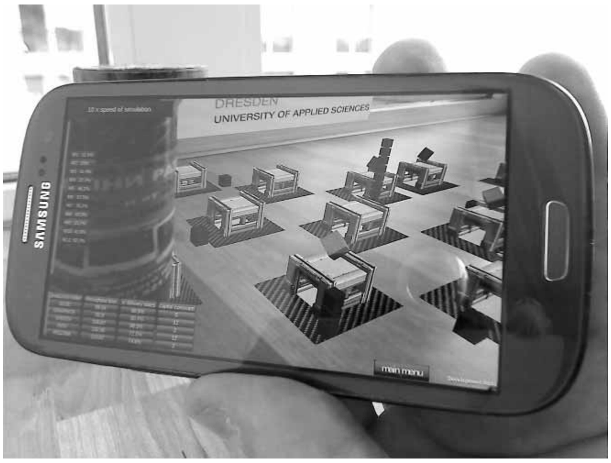


Figure 2: Visitor’s view of the AR model factory.

The following three sections describe the components of the system and their functionality in detail.

3 Event-based Simulation of the Model Factory

The model factory is a job shop system (see [7]): An infinite stream of production orders enters the factory. Each production order requires a sequence of operations to be processed. Each operation is performed by a specific machine and requires a certain processing time. Each production order belongs to a certain product type assigned to a specific routing (i.e., a specific sequence in which the production order seizes the machines). A machine can process only one production order at a time, and one production order can only seize one machine at a time. While a machine is processing an order, other incoming orders enqueue in front of the machine.

Every machine has its own queue with unlimited capacity. After an operation is completed, the production order moves on to the next machine. In our model, only the transportation distance influences the required transportation time. Transportation resources are not part of the model. For demonstrating the model factory, we applied a production system consisting of eleven machines and producing five types of products (see [8] und [9]). Figure 3 pictures the production system.

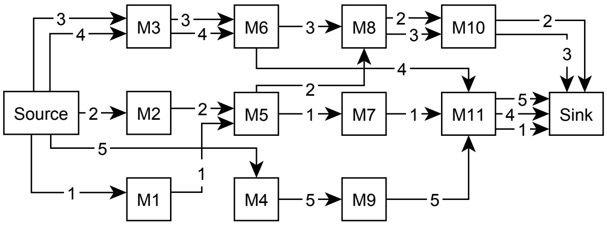


Figure 3: Production system containing eleven machines and the routings of five different products.

Typically, production planning and control pursues four objectives, namely [10]

- adherence to delivery dates,
- short throughput times,
- high utilization of the machines, and
- low level of work in process.

Four key performance indicators (KPI) operationalize these objectives. The KPIs are calculated and updated during the simulation.

A number of parameters is necessary to describe the model factory:

- number of machines,
- distribution of interarrival times,
- product types, their probability, and their specific routings (sequences of operations processed on certain machines and requiring certain processing times),
- velocity of transport, and
- control strategy applied to select orders from waiting queues.

An administrator writes the values of these parameters to an XML file. This configuration file serves as input for generating an event-based simulation model.

A variety of simulation systems is able to simulate job shop systems. We chose DESMO-J (see [11] and [12]), version 2.5.1c, developed by the Department of Computer Science of Hamburg University. DESMO-J is an open-source object-oriented simulation framework written in JAVA™.

JAVA™ is also used to realize the simulation models itself. DESMO-J supports process-oriented and event-oriented modeling [11]. We chose the latter because of its global view, its simplicity, and much lower computational costs.

While data-driven model generation is a well-known technique in simulation [13], there is one aspect that is worth mentioning: For sake of simplicity, the configuration file does not contain information about the physical location of the machines on the shop floor or a geometric description of the network of transportation routes. Hence, the factory layout has to be created automatically: An algorithm for layered drawing of directed acyclic graphs places the virtual machines relative to each other on the factory floor. The machines are the nodes of the graph, the transportation routes between consecutive machines are the directed edges of the graph. The algorithm aims at minimizing the number of crossings while keeping the factory layout as small as possible (see [14]). In order to achieve both, the algorithm sorts the machines topologically and assigns them to virtual layers (see [15]). Afterwards, the algorithm heuristically exchanges the machines at each layer to minimize the number of crossings of transportation routes (see [16]). All transportation routes are either parallel or perpendicular to each other and they may have perpendicular bends.

The model generator writes the coordinates of the machines and the transportation routes to the database (see Section 5). This way, the AR app can access the geometric information regarding the factory layout.

Once the model generation is completed, the simulation starts and runs until manually terminated. Visitors can not only view the operating model factory but also change some parameters at runtime:

One parameter is the utilization of the production indicated by the utilization of the bottleneck machine. The system limits the utilization to 85 % because the factory and the queues should not overflow with production orders. When a visitor enters a new utilization, the system calculates an appropriate interarrival time of the production orders applying the probabilities of the product types and the routings. Thus, the system generates production orders more or less often, and after a certain time the utilization converges to the required value. [17] gives an example of the above calculations.

The visitor may also change the product mix. She or he can choose one of three options:

- the initial distribution read from the configuration,
- a uniform distribution (all product types appear with the same probability), or
- a distribution where the next product type is twice as probable as the current one, i.e., if there are product types $i = 1, \dots, n$, the next arriving order will require product i with probability $p_i = 2^{i-1}/(2^n - 1)$.

Furthermore, the visitor can choose from several queueing strategies [18]:

- first come, first served,
- shortest processing time first,
- shortest slack time first, and
- highest value first.

All queues will be re-sorted when the global queueing strategy changes. The slack time and the values of the production orders are calculated and updated during the course of the simulation.

When a visitor changes a parameter value, the model factory needs time until it reaches a steady state again. The method ‘crossing of the means’ (see [19]) indicates when this is the case. Until then, visitors cannot change parameter values further. This way, the visitor can experience the effects of her or his change on the KPIs representing the four objectives of production planning and control.

4 The Composition and the Functionality of the AR Mobile App

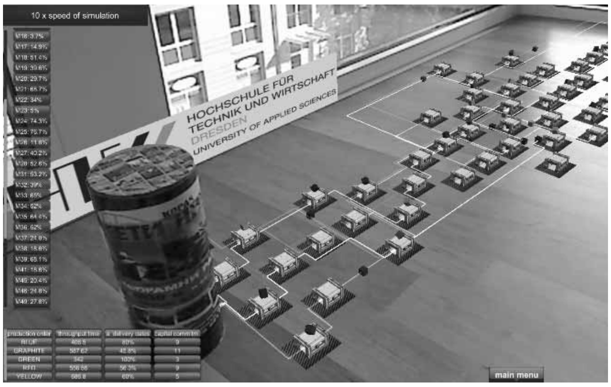
Two frameworks support the visualization of the AR model factory, namely Unity and Vuforia. Unity is a runtime and development environment for 3D visualization and interaction (see [20] and [21]). It is a product of Unity Technologies, Inc. We applied free Unity Personal, Version 5.2.2. Vuforia is a software development kit for mobile devices that enables the creation of AR applications (see [22] and [23]). Vuforia was developed by the Qualcomm Developer Network and is now product of the Parametric Technology Corporation, Inc. We applied the non-commercial version Vuforia 5.5.9.

The 3D visualization of the model factory in Unity is straightforward: The machines are ready-made visual objects. The production orders are simple cubes differently colored in order to distinguish their routings/products. If necessary, they are stacked in front of the machines in order to resemble waiting queues.

If a stack becomes too high, its order cubes are scaled down proportionally. If a machine is processing a production order, a spinning cube above the machine indicates that the machine is working.

The transportation routes are drawn as parallel or perpendicular lines with perpendicular bends if necessary. The transportation routes are colored according to the color of the routings/products. The data for the visualization stems from the simulation (see Section 3). Unity calculates the movement of the production orders on the transportation routes, which results in a visualization of a smooth flow of the orders along the routes. Figure 4 shows the AR model factory containing machines, waiting queues, production orders, transportation routes etc.

Unity also provides control widgets. We applied widgets to change the utilization of the model factory, to change the product mix, and to change the queueing strategies (see Section 3). The widgets do not appear inside the 3D visualization of the factory but at the 2D graphical user interface (GUI) of the mobile app. Furthermore, the GUI displays the current values of the objective variables. This way, the visitor can observe how the change of the parameter values effect the objectives. Figure 4 shows the widgets and the display of the values of the four KPIs mentioned in Section 3.



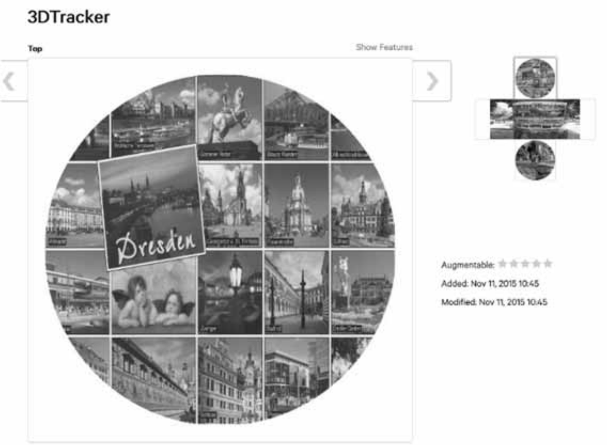
production order	throughput time	ø delivery dates	capital commitm
BLUE	408.5	80%	9
GRAPHITE	587.67	45.8%	11
GREEN	342	100%	3
RED	556.56	56.3%	9
YELLOW	686.8	60%	5

Figure 4: Screenshot of the AR model factory; values of the objective variables at the bottom left corner.

All visitors watching the model factory via their mobile devices see the same factory at the same state. After one of them changes parameter values, all controls on all mobile devices become inactive until the model factory reaches a steady state again which usually takes less than two minutes. This way, the visitors can observe the undistorted effect of the change of the parameter values.

When we developed the AR app, we decided on marker-based augmented reality. The app knows the marker beforehand, and thus, the app identifies the marker more easily. In marker-less AR settings, the app has to identify certain objects, like faces, without knowing every object beforehand, which is difficult and therefore computationally expensive [24].

We decided on a cylindrical marker pasted up with parts of picture postcards (one of them from Dresden, Germany). Figure 1 shows the physical marker in the top right corner. Next, we defined a corresponding virtual marker at the website of Vuforia [25]. Figure 5 shows the marker's definition.


3DTracker

Top

Show Features

Augmentable: ☆☆☆☆☆

Added: Nov 11, 2015 10:45

Modified: Nov 11, 2015 10:45

Figure 5: Definition of the marker: top view and overview over the complete surface.

The virtual marker resembles the physical marker. Its surface shows exactly the same pictures at exactly the same place. We downloaded a package from the Vuforia website, which contains the logical/virtual marker and a virtual AR camera. We imported this package in Unity and placed the logical/virtual marker inside the model factory, which already exists as rudimentary 3D model in Unity. We placed the logical/virtual marker at the entry of the model factory to represent the source of the production orders. Figure 6 shows the logical/virtual marker in Unity.

In order to establish an AR scenario, we need to insert a 3D visualization of a virtual world into a real-time video stream of a real-world scene. The camera of the mobile device provides the necessary video stream. In our case, the scene consists of a table with the physical marker standing on top of it. The AR app receives the video stream, identifies the marker, calculates its position and orientation, and provides an origin of coordinates as reference.

With the virtual camera, the Vuforia-part of the app then projects the virtual Unity model on the video stream, i.e. on the table (and the marker). The app finally delivers the augmented video stream to the visitor on the screen of her or his mobile device in real-time. The visitor gets the impression that she or he is watching the model factory on top of the table through the mobile device.

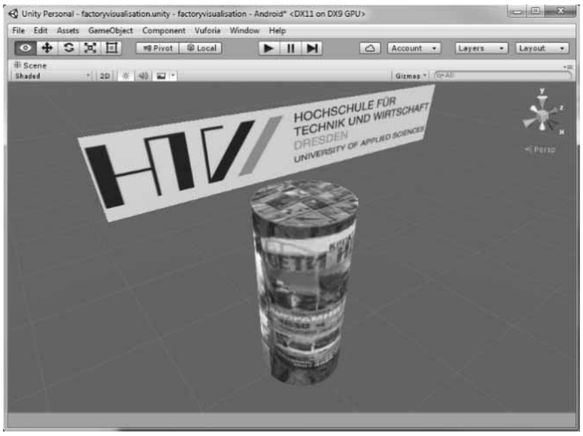

Unity Personal - factoryvisualisation.unity - factoryvisualisation - Android* <DX11 on DX9 GPU>
File Edit Assets GameObject Component Vuforia Window Help
Pivot
Local
Account
Layers
Layout
Scene
Shaded
2D
Gizmos
Persp
HOCHSCHULE FÜR TECHNIK UND WIRTSCHAFT DRESDEN UNIVERSITY OF APPLIED SCIENCES

Figure 6: Representation of the virtual marker in the Unity framework.

One major problem occurred when Vuforia overlaid the real-time picture with the 3D visualization: The visualization alternated abruptly and noticeably between two positions in consecutive resulting pictures. The reason for the undesirable blurring is a picture downscaling effect: For faster real-time pattern recognition of the marker in the real-world picture, Vuforia scales the input picture down to a size of 640×480 pixels.

The remaining pixel array serves the detection of the edges of the marker. If the mobile device and its camera move only slightly, their position will repeatedly alter marginally in relation to the physical tracker and the detected edges of the marker in the picture will jump between adjacent pixels. Scaled up again, the distance between the marker's edges detected in one picture differs considerably from the edges detected in the next picture. Thus, Vuforia overlays the 3D visualization on the real-world pictures at different positions in consecutive pictures. Consequently, the 3D visualization jumps noticeably from one resulting picture to the next. We solved the problem by using the moving averages of the last five detected positions and orientations of the marker in the Vuforia software module when overlaying the real-world picture with the 3D visualization. This proved to be sufficient to avoid jumps of the 3D model in consecutive resulting pictures.

After compiling the app in Unity, it can be exported as Android Package (APK) for the Android operating system or as XCode for iOS.

5 The Database for the Communication between Simulation and Mobile App

The database is stored on the notebook and managed by MySQL. Both the simulator and the app are connected to the database via JDBC.

The database serves as interface for the bidirectional communication between the simulation and the AR app. In brief, the simulation writes master data and events to the database while the mobile app reads the master data and the events from the database and applies them to the visualization. On the reverse channel, the mobile app writes control data to the database while the simulation reads the data from the database and adopts the simulation accordingly.

Figure 7 shows a simplified data model in crow's foot notation (see [26]). Initially, the simulation model generator writes master data to the database: machines and their locations, transportation routes with their starts, stops, and bends. As a simulation run proceeds, the relevant events of the order lifecycle are written to the database: entering the factory, enqueuing in a waiting queue, seizing a machine, releasing a machine, and leaving the factory. Moreover, the simulation writes KPIs to the database. The simulation adds timestamps to all records. Database triggers aggregate, simplify, and prepare all data relevant for the visualization and distribute the data to small tables containing only very few records. Reading these tables is very fast.

The mobile app reads the master data at the beginning of the simulation and creates the 3D visualization of the factory layout. Then, the mobile app periodically polls events from the database. Every period comprises a few frames. According to the events, the mobile app updates the 3D visualization of the production orders. As mentioned in Section 4, the only data not provided by the database are the locations of the production orders on the transportation routes during transport. The mobile app continuously calculates these locations in relation to the length of the routes and the given velocity.

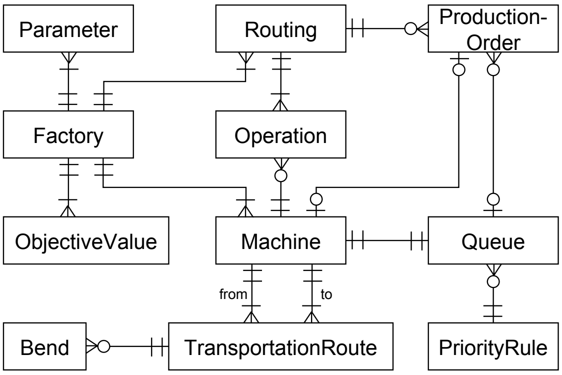


Figure 7: Simplified data model.

When a visitor changes the value of a control parameter, the mobile app writes the change and a timestamp to the database. All connected apps read the new parameter values from the database, update the parameter values in their graphical user interface, and disable the controls until the simulation reaches a steady state again. Thus, no visitor can change parameter values until then. If another visitor changes the parameter values in the meantime, the database will discard the changes because of the later timestamp. In parallel, the simulation also reads the new parameter values and adopts its behavior accordingly. Thereupon, the simulation writes newly occurring events to the database and updates the KPIs. The mobile app reads the updated KPIs from the database and displays them to the visitor. When the simulation reaches a steady state again, the simulation writes the information to the database and the mobile app reads the information and enables the controls.

6 Watching and Operating the Model Factory through a Personal Mobile Device

When exhibiting the AR model factory, we provide mobile devices, so visitors can watch the factory without their personal mobile devices. However, there may not be enough mobile devices for all visitors and some visitors might want to watch the factory through their personal devices.

In order to install the AR app on his or her own device, the visitor has to scan a QR code printed on a poster provided at the booth. The QR code encodes an URL referring to a website located on the local notebook/PC at the booth. Using the URL, visitors can download the installation files of the mobile app for different versions of different operating systems.

Once the app has been installed, it can be used immediately to view and control the AR model factory.

7 Recent and Future Use of the AR Model Factory

We already exhibited the AR model factory twice at the career-start fair (KarriereStart) in Dresden, several times during open houses at Dresden University of Applied Sciences, and during Long Night of Sciences (Lange Nacht der Wissenschaften) in Dresden. Visitors were very interested in the model factory. More than 150 of them watched the model factory ‘through’ our prepared tablet computers on the table; more than 40 of them installed the mobile app on their own mobile devices. All visitors changed at least the utilization of the factory and watched the waiting queues lengthen or shorten. The feedback was positive without exception. The most frequent question asked by visitors was if we could also visualize and control their real-life factories in AR in real-time.

The exhibition scheduled next is the upcoming Long Night of Sciences in Dresden. When introducing Dresden University of Applied Sciences at high schools, we will also exhibit the AR model factory. We also plan to present the model factory during poster sessions at scientific conferences.

8 Summary and Outlook

We described the development of an AR model factory. The overall goal was to build an eye-catcher we can apply at trade fairs, exhibitions, conferences, open houses etc. We achieved the goal in full by building the AR model factory. Evidently, the model factory stimulates the visitors to approach the booth and to gather information of the courses of study offered at our faculties/universities. Moreover, the model factory has a strong relation to a specific domain of business informatics, namely production planning and control: We can use the model factory to demonstrate some principles of production planning and control. The eye-catcher is not physical, but some of its parts are moving, and it sends optical and acoustic signals when certain events occur in the factory. In essence, the visitor can watch the model factory ‘through’ her or his smartphone. The model factory seems to stand on top of a physically empty table and it is visibly processing production orders of different types.

Visitors may manipulate the behavior of the model factory by changing the utilization, the product mix, and the queueing strategy. The changes result in different values of the KPIs (like adherence to delivery dates or work in process), which the visitor can also monitor.

At a next stage of extension, the visitor could interact with the mobile app via speech and gestures but speech and gesture recognition might slow down the visualization. If the simulation and the database run on a remote server, the notebook/PC will become obsolete at the booth and we will need the marker only. The internet connection should be broad and stable for this scenario, which is often not the case at trade fairs. The visualization could display the number of waiting orders above the queues. Furthermore, the development of the KPIs could be visualized as time lines in diagrams. A printout of the factory layout and physical building blocks representing the machines could serve as a physical model of the factory. This way, the model factory would interlace reality and AR even more and it would more closely resemble a physical model factory.

References

[1] März L, Krug W, Rose O, Weigert G. *Simulation und Optimierung in Produktion und Logistik*. Berlin: Springer; 2011. 220 p.

[2] Bickel S, Yildiz M, Blessing B, Kriz WC, Oslowski G, Eiselen T. The e-Beer game: an online simulation game for the training of supply chain management. In: Mayer I, Mastik H. editors: *Organizing and Learning Through Gaming and Simulation. Procedings of ISAGA 2007*; 2007 Jul; Nijmegen. Delft: Eburon. 39-46.

[3] http://www.fischertechnik.de/en/Home.aspx

[4] Mayer F, Pantförder D, Diedrich C, Vogel-Heuser B. *Deutschlandweiter I4.0-Demonstrator*. Lehrstuhl für Automatisierung und Informationssysteme, Technische Universität München, 2013.

[5] http://www.festo-didactic.com

[6] http://www.arena2036.de

[7] Blazewicz J, Domschke W, Pesch E. The Job Shop Scheduling Problem: Conventional and New Solution Techniques. *European Journal of Operational Research.* 1996; 93: 1-33.

[8] Schulz R. *Parallele und Verteilte Simulation bei der Steuerung komplexer Produktionssysteme*. Technische Universität Ilmenau; 2002.

[9] Bube O. *Architektur und Implementierung von Prozeßidentifikatoren*. Technische Hochschule Ilmenau; 1993.

[10] Buzacott JA, Corsten H, Gössinger R, Schneider, HM. *Production Planning and Control: Basics and Concepts.* München: Oldenbourg; 2012. 246 p.

[11] Page B, Kreutzer W. *The Java Simulation Handbook: Simulating Discrete Event Systems with UML and Java*. Aachen: Shaker; 2005. 516 p.

[12] Page B, Lechler T, Claasen S. *Objektorientierte Simulation in Java mit dem Framework DESMO-J*. Books on Demand; 2000. 197 p.

[13] Bergmann S. *Automatische Generierung adaptiver Modelle zur Simulation von Produktionssystemen*. Technische Universität Ilmenau; 2013.

[14] Battista GD, Eades P. *Graph Drawing: Algorithms for the Visualization of Graphs*. Upper Saddle River, NJ: Prentice-Hall, 1998. 397 p.

[15] Coffman EG, Graham RL. *Optimal Scheduling for Two-Processor Systems. Acta Informatica*. 1972; 1: 200–213.

[16] Barth W, Jünger M, Mutzel P. *Simple and Efficient Bilayer Cross Counting. Journal of Graph Algorithms and Applications*. 2004; 8(2): 179-194.

[17] Munkelt T. *Potenzial Bayes'scher Netze zur Unterstützung der Produktionsplanung und -steuerung*. Technische Universität Ilmenau; 2009.

[18] Shtub A, Karni R. *ERP: The Dynamics of Supply Chain and Process Management*, New York: Springer; 2010. 281 p.

[19] Mahajan PS, Ingalls RG. Evaluation of Methods Used to Detect Warm-up Period in Steady State Simulation*. In: Ingalls, R. G., editors: Proceedings of the Winter Simulation Conference;* 2004 Dec; Washington, DC. 663-671.

[20] Goldstein W. *Unity Game Development Essentials.* Birmingham: Packt; 2009. 316 p.

[21] Chittesh J. *Das Unity-Buch: 2D- und 3D-Spiele entwickeln mit Unity 5.* Heidelberg: dpunkt; 2015. 512 p.

[22] Kwik F, Bahana R. *Using Augmented Reality to Enhance Aetherpet, a Prototype of a Social Game.* Procedia Computer Science. 2015; 59: 282-290.

[23] Imbert N, Vignat F, Kaewrat C, Boonbrahm P. *Adding Physical Properties to 3D Models in Augmented Reality for Realistic Interactions Experiments.* International Conference on Virtual and Augmented Reality in Education; 2013 Nov; Puerto de la Cruz. 364-369.

[24] Chi HL, Kang SH J, Wang X. *Research trends and opportunities of augmented reality applications in architecture, engineering, and construction.* Automation in Construction. 2013; 33: 116-122.

[25] https://developer.vuforia.com/target-manager

[26] Rob P, Coronel C. *Database Systems: Design, Implementation & Management*. 8th ed. London: Cengage Learning EMEA; 2008. 704 p.

An Introduction to Discrete-Event Modeling and Simulation

Oliver Ullrich[1]*, Daniel Lückerath[2]

[1]School of Computing and Information Sciences, Florida International University, ECS 243C, 11200 SW 8th St, Miami FL-33199, **oullrich@cs.fiu.edu*

[2]Institut für Informatik, Universität zu Köln, Weyertal 121, 50931 Köln, Germany

SNE 27(1), 2017, 9-16, DOI: 10.11128/sne.27.on.10362
Received: October 10, 2016
Accepted: December 10, 2016
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna,
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Especially suitable for the modeling and simulation of technical systems in a wider sense, discrete-event simulation is one of the most important and most versatile tools of the craft. Both students and practitioners should be familiar with its principles and mechanics, in particular if they are interested to look beyond their favorite modeling tool's GUI.

This paper presents a short tutorial on modeling and simulation techniques, with a focus on discrete-event simulation. After an introduction to the subject matter, some event-oriented modeling fundamentals are discussed, followed by a description of simulation execution principles. A slim software framework is introduced aimed at simplifying model building and evaluation, followed by the presentation of a small sample of recently completed discrete-event simulation studies. The paper concludes with a short summary of the lessons learned and some pointers to additional literature.

Introduction

Decades after its inception (see [19]), the discrete-event modeling and simulation method is still a mainstay of the trade, with hundreds of discrete-event simulation models being built, executed, and described (for a small selection of recently completed studies see section 4).

Discrete-event based methods are especially suitable if the examined system's behavior can be described as a set of stochastically influenced, connected components, changing their states at discrete points in time. A discrete-event simulation model represents such a system as a set of connected, attributed entities, executing, and communicating by, events and activities.

Common applications range from communications to transportation, and especially cover the simulation of technical systems in a wider sense. It is less suitable in areas like astronomy, meteorology, or materials sciences, while social scientists often prefer agent-based modeling for their simulation experiments.

While discrete-event simulation is easily understood, it is also very expressive and powerful. Single researchers, students, or practitioners can create high-validity models and evaluate their behavior. Discrete-event simulation includes three worldviews: activity-oriented, process-oriented, and event-oriented simulation (for detailed information see e.g. [5], pp. 88-108). While these views each emphasize different model components to represent the investigated system, they all use the same basic modeling principles and can easily be translated into another.

This paper presents an introduction to modeling and simulation techniques, with a focus on discrete-event simulation. It is especially addressed to students of the craft, and to practitioners who might want to look beyond their usual modeling tool's GUI. In addition to describing the modeling process, the execution of simulation models, and a number of typical academic and real-world applications, the paper also includes a description of an open-source Java framework which encapsulates and hides the event-oriented method's technical details, and provides a slim API for modelers to start implementing a model without having to be concerned with the underlying mechanics. The framework can be used (and has been used) for both simple experiments and full blown case studies.

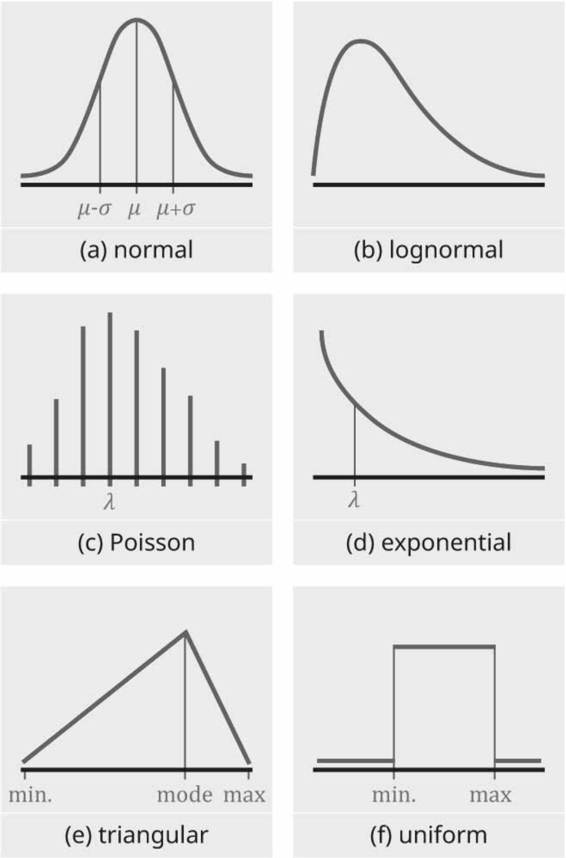
The paper continues with an introduction of basic modeling methods with a focus on event-oriented concepts (section 1), including entities, events, activities, as well as input distributions and their modeling, followed by a description of fundamental techniques of model execution (section 2), both for traditional event-oriented simulation and its object-oriented extension. Building up on this, the paper then introduces a slim event-oriented simulation framework (section 3) designed to allow the user to concentrate on modeling aspects and to isolate them from technical housekeeping work. The penultimate section (section 4) presents a sample of current or recently concluded case studies in a wide variety of application fields, including communications, disaster mitigation, health care, logistics, supply chain management, and transportation. The paper concludes with a short summary of the lessons learned and some recommendations for further reading (section 5).

1 Event-oriented Modeling

The modeling process starts with the examination and analysis of the real-world system to be modeled. It is decomposed into interacting components which perform processes, execute and generate events and interact by passing messages. The decision has to be made what components and processes should be part of the model, what should be parametrized, and what should be left out completely. Generally, if an entity or a process has a strong influence on the core behavior of the model, it should be incorporated. If the influence is weak, it might be wiser to parametrize it to avoid creating overcomplexity.

Discrete-event modeling is suitable for systems that can be described as a set of interrelated entities which only change their state (and subsequently the system's state) at discrete points in time as a result of their own behavior or of the behavior of other entities. Entities describe components of the system under investigation which have to be modeled explicitly in order to represent the system behavior relevant to the simulation study. Each entity possesses individual attributes whose values describe their current state and which affect their behavior. The potential behavior of an entity is modeled as a set of activities, i.e. a sequence of actions during some time period which may result in state changes of the acting entity as well as other entities of the model. The point in simulation time at which such a state change occurs is called an event. Subsequently, activities are always framed by events. The precise relationship between specific events and activities has to be defined by the modeler based on the goals of the simulation study. In particular, this includes the specification of the durations of activities, which can be modeled as deterministic as well as based on stochastically influenced parameters. The latter case is usually modeled using drawings from random distributions, whose type and parameters are acquired by analyzing relevant input data (e.g. using Microsoft Excel).

In discrete modeling some distribution types (see figure 1) are especially useful: Independent machine processing times, e.g. for the tooling of a component, can often be approximated by applying a normal distribution $\aleph(\mu,\sigma^2)$ with an average of μ and a standard deviation σ, or by using a log-normal distribution $\ln\aleph(\mu,\sigma^2)$. Arrival rates, e.g. of cars entering a freeway segment, are usually modeled using a discrete Poisson distribution $\mathrm{pois}(\lambda)$ with an average number λ of events per interval. The time between two independent arrival events is often approximated with an exponential distribution $\exp(\mu)$ with an average interarrival time of $\mu = 1/\lambda$, λ again being the average number of events per interval (see [5], pg. 248).



(a) normal (b) lognormal (c) Poisson (d) exponential (e) triangular (f) uniform

Figure 1: Distribution types.

If the available data is insufficient to determine a distribution's type and parameters, insight might be gained by interviewing local domain experts. If an expert is confident that a stochastically influenced process takes a time "usually in the vicinity of *mode* units, never less that *min*, never more than *max* units," the process duration can be modeled as a drawing from a triangular distribution $\mathrm{tri}(mode, min, max)$ with the parameters *min*, *max*, and *mode*. If available expert knowledge is as imprecise as "somewhere between *min* and *max*," the modeler has to make due with a uniform distribution $\mathrm{U}(min, max)$ with the parameters *min* and *max*.

Because the model state only changes at the occurrence of events, it follows that it stays unchanged between two subsequent events and that in general an activity can't be influenced by an event which occurred after the activity has already started.[1] The event-oriented worldview takes advantage of this by representing the passage of time in the real-world system using variable time steps for the incrementation of simulation time, i.e. the simulation time jumps from the occurrence of one event to the occurrence of the next event. The event-oriented simulation process can thus be understood as a sequence of snapshots of the model state, starting with the earliest snapshot (e.g. at simulation time $t_s = 0$) and ending once no more snapshots are generated. The first snapshot in the sequence has a special status: It has to be defined by the modeler and is used to initialize the states of all model entities. Furthermore, it is used to schedule exogenous events, i.e. events whose origins lie outside the boundaries of the model (in contrast to endogenous events, whose origins lie inside the boundaries of the model). Often, the practitioner's goal is not to examine a model's regular behavior, but to evaluate its reaction to specific disruptions or disturbances (e.g. the blockade of a track segment or a machine failure). In such a case one does not want to rely on a disruption occurring randomly, especially if these are very rare. Therefore, disruptions are systematically injected into the model as exogenous events with a specific time stamp during the first simulation snapshot. When the simulation is started, it executes the routine events, until it reaches the exogenous event. The model's state is changed accordingly and the model logic designed to cope with a disruption springs into action; its effectiveness can now be examined systematically.

A commonly used tool to plan and visualize event-oriented models is the event activity chain (see figure 2). Here, events are represented as hexagons, framing activities which are represented as round cornered rectangles. Conditions to be fulfilled, or a decision on what path to follow, can be represented by *or* ($\vee$), *exclusive or (xor)*, or *and* ($\wedge$) connections. The chain's elements can further be attributed by distributions and other parameters.

2 Event-oriented Model Execution

Once the components of a simulation model are identified, including their attributes, relations, activities, and events, the model is implemented as software and subsequently executed. Using object-oriented programming languages, model entities can be implemented one to one as objects, with entity types being realized as classes.

The data structure used to represent events has to include fields for: the event type, e.g. START_SIMULATION, BEGIN_PASSENGER_EXCHANGE, RELEASE_WORKSTATION; the time stamp, usually stored as an integer value, with one increment constituting the smallest representable time increment, e.g. one second or one millisecond; and additional information depending on the model.

These simulation events are managed by the Future Event List (FEL). This data structure is usually implemented as a priority queue, an abstract data type which orders elements according to a key, in this case to the time stamp. Amongst other functions, the FEL provides interfaces to find and remove the element with the smallest key, insert new elements at the right position in the FEL, and test for emptiness. Usually, the dequeuing order of events with identical time stamps is not guaranteed, therefore making it more difficult to represent concurrent access to shared resources. If such functionality is desired, the events have to include an additional attribute, e.g. a general index, which can be used as secondary sorting key.

During the simulation run the simulation engine repeatedly retrieves (and thus removes) the event with the minimum time time stamp value from the FEL and executes it. In the course of the execution the simulation time is set to the currently executed event's time stamp, the model's state variables are updated, and follow-up events are generated

[1] It is possible to deviate from this assumption in individual cases, if the simulation framework used to implement the model provides methods and data structures for the rescheduling of already scheduled events, thus possibly aborting an activity that has already started.

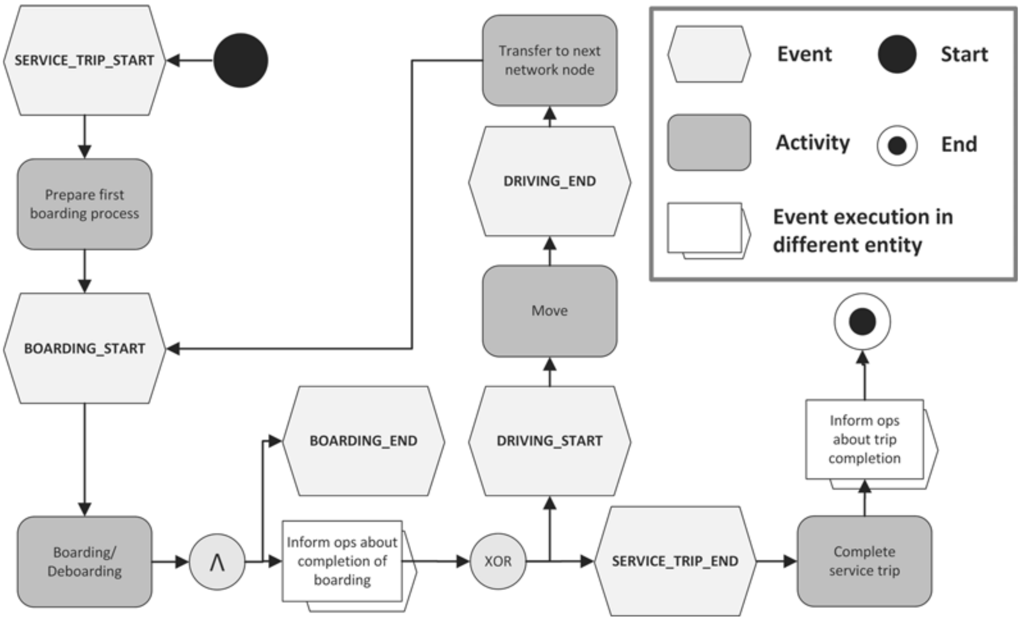

SERVICE_TRIP_START
Prepare first boarding process
BOARDING_START
Boarding/ Deboarding
Λ
BOARDING_END
Inform ops about completion of boarding
XOR
DRIVING_START
Move
DRIVING_END
Transfer to next network node
SERVICE_TRIP_END
Complete service trip
Inform ops about trip completion
Event
Start
Activity
End
Event execution in different entity

Figure 2: Example for event-activity-chain notation depicting a simplified model of a transit vehicle executing a service trip.

and inserted into the FEL (e.g. during the execution of BEGIN_PASSENGER_EXCHANGE an event END_PASSENGER_EXCHANGE might be scheduled 23 seconds into the model's future). The simulation therefore jumps from time stamp to time stamp, and ignores the time in between two subsequent events. Stochastic elements, e.g. processing durations or interarrival times, are drawn from random distributions, usually by utilizing a library providing a large number of distribution types. How to efficiently generate these distributions, and how to ensure their unpredictability and randomness, is a research area by itself (see [10]), and is well beyond the scope of this introduction.

Using the described data structures and concepts, the simulation run is executed by iterating through a simple loop (see figure 3): (1) Start the run, set the simulation time to zero, send a *START_SIMULATION* event when appropriate; (2) get the event e_i with the minimum time stamp from the FEL, set the simulation time to e_i's time stamp; (3) execute event e_i according to its type, thereby updating the model state and enqueuing follow-up events; (4) check the stop conditions, which usually include at least a check whether the FEL is empty. If the stop conditions are fulfilled, end the simulation run by continuing with step (5). If not, repeat the loop from step (2) by dequeueing the next event e_{i+1}. (5) Prepare statistics and evaluate results.

In the object-oriented extension (see figure 4) the event data structure additionally contains a reference to the source entity and a list of entities which are the event's addressees. The central event handling routine is replaced by a dispatcher which calls the addressees' event handling routines. If one event includes more than one addressee, the order of these calls usually cannot be guaranteed.

With potentially hundreds or thousands of events in the queue at the same time, it is obvious that the performance of the priority queue implementation is an important factor in the method's overall performance. A number of data structure is commonly used to implement the priority queue, including hashing structures, heaps, binary trees, or (in small models) linked lists. For a more detailed discussion of priority queue data structures see [16].

3 An Event-oriented Simulation Framework

Surfside[2] is a slim Java framework for event-oriented simulation. It has been used in a number of applications (see [12], [13], [15], [25]), mainly in the area of computational transportation science. *Surfside* is designed to

[2] available under an open source license at www.oullr.de/surfside

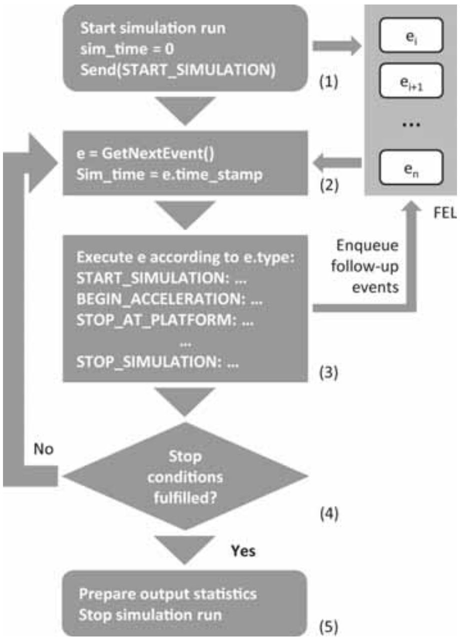


Figure 3: Basic event-oriented simulation loop.

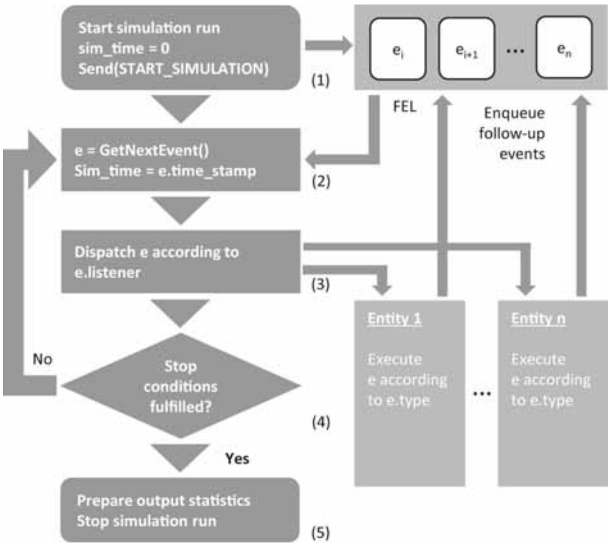


Figure 4: Object-oriented extension of the event-oriented simulation loop.

be easily understood, and to help modelers to concentrate on the actual modeling task by isolating them from technicalities.

Simulation events are represented by instances of the class *Event*. Each instance includes an event type value, a time stamp, a reference to the event's generating entity, and a list of event listeners or addressees, i.e. objects implementing the interface *EventHandler*. The class *Event* can be extended to include further application-specific data.

Event types are administrated by the class *EventTypeManager*, which prevents dublicated event type values and provides centralized access to all registered event types. A set of standard event types, defined in the class *EventType*, is added to the *EventTypeManager* by default. To use custom event types the user has to create her own event types and add them to the *EventTypeManager*.

Entity types are realized as classes implementing the interface *EventHandler*. This interface demands a method *handleEvent(Event)* which consumes simulation events addressed to the entity, evaluates event data, changes the model state, and generates follow-up events by calling *Simulation.addEvent(Event)*. The interface also demands a method *reset()* to reset the entity's state in between simulation runs, and some other minor methods.

The abstract class *Simulation* encapsulates the functionality to prepare, execute, and evaluate a number of simulation runs and is usually extended to be the simulation application's main class. It is implemented as a Singleton and itself implements the *EventHandler* interface, and can thus function as a conceptual model entity. If all simulation events are handled by the class extending *Simulation*, the model followed the classic event-oriented paradigm without its object-oriented extension.

The user's code takes control in an extended constructor, utilizing calls to *addEntity(EventHandler ...)* to add one or more entities to the model. The number of simulation runs to be executed is set via *setNumberOfRuns(int)*. The user's code then calls *start()*, thereby commencing the execution of the given number of simulation runs. The user also has to implement the abstract methods *prepareNextRun()*, *eventPostProcessing(Event)* and *evaluateRun()*, which are used to reset the model in between simulation runs, evaluate and collect data after processing a single event, and to evaluate data collected after finishing a run, respectively. At the start of each simulation run, an event of type *SIMULATION_START* is automatically generated and sent to the class extending Simulation, at the conclusion of each run, if the user did not stop the simulation herself, an event of type *SIMULATION_END* is automatically sent. The class also manages the FEL, to which events can be added by using *addEvent(Event)*.

Specific events can be removed from the list by calling *removeEvent(Event)*. The framework automatically manages the FEL mechanics encapsulated in the class *FutureEventList*. This class can be extended, thereby allowing experiments with FEL data structures themselves.

Surfside provides a number of helper classes for input distribution generation, output logging, etc: The class *Distribution* encapsulates parametrized random distributions for single time periods with homogeneous parameters, supporting uniform, triangular, normal, lognormal, exponential, and poisson distributions. The class *CompositeDistribution* encapsulates functionality to model processes whose distributions and parameters change over time, by allowing the assemblage of mutlipe *Distribution* instances with inhomogeneous parameters and different time periods. The class *PropertiesManager* administrates the simulation parameters and offers functionality to read parameters from the command line as well as from XML files. The classes *Column* and *Exporter* help to collect and write output data. The latter is an interface defining the minimum functionality for storing simulation data, e.g. *add(Column)* and *log()*, which should be implemented by custom data export classes. A simple implementation for writing data to a Microsoft Excel spreadsheet based on a user defined template using the JExcelAPI libary is already provided by the class *JExcelAPIExporter*.

4 Current Applications

Presented here is a small sample of current or recently concluded discrete-event simulation case studies in a wide variety of application fields, collected to give an impression of the method's flexibility and utility. Some applications show small, very specific models, constructed and implemented by single practitioners or a small team, while others consist of large multi-person projects.

Communications: Artuso and Christiansen (see [3]) examine inter-cell interferences in long term evolution (LTE) cellular networks. As it is very costly to assess possible solutions in the field, network operators are interested in the evaluation of potential remedies in simulated scenarios. Artuso and Christiansen find that, by applying certain techniques, improvements from 25% to 40% in relation to the baseline scenario can be achieved. Cetinkaya et al. (see [6]) analyze the effects of perturbations on resilient communication networks. They use discrete-event simulation to inject disturbances into a representation of the USA's Sprint backbone network, thus assessing impacts on performance and dependability.

Disaster mitigation: Kuchel (see [11]) examines the impacts of civil unrest and violence on large-scale evacuation operations. He provides a model to analyze and evaluate scenarios with the goal of reducing evacuation time, and to identify limiting factors and choke points for given local transportation networks. Suk Na and Banerjee (see [23]) combine agent-based and discrete-event simulation to provide a decision support system for network evacuation strategies in case of natural disasters. Their model includes an embedded geographic information system, and allows to evaluate possible shelter positions, vehicle types, and evacuation priorities. They examine a number of scenarios based on an assumed natural disaster in Galveston, Texas.

Health care: Allen et al. (see [2]) examine the rescheduling of medical surgery schedules as a consequence of disruptions during the operational day. Their discrete-event model focuses on the impacts of rescheduling measures on hospital personnel, especially on further delays resulting from delayed communications regarding the updated schedule. Mayorga et al. (see [18]) construct a simulation model to assess the health and economic impacts of various smoking cessation techniques. Their model represents 100,000 smokers receiving one of five alternative smoking cessations interventions, including several medications, nicotine replacement therapy, and counseling. The results indicate that while all treatments have a positive average effect on the participants' quality of life, a small long-term preference for a certain pharmaceutical treatment can be identified. Pooya et al. (see [20]) use discrete-event simulation to analyze and assess the reliability and safety of the radiation therapy care delivery process, focusing on possible optimization by quality assurance protocols. They find that "incidents" caused by mistakes in the organ delineation and dose calculation steps are most costly, and can be reduced by improved quality assurance procedures.

Logistics and supply chain management: Abbot and Marinov (see [1]) examine alternative designs for railroad interchange yards, which facilitate the exchange of rolling stock between Britain's planned "High Speed Two" railway and conventional networks. Their model consists of a number of sub-models, representing e.g. locomotive change, brake inspection, and the train arrival process. They find that under the given yards designs high-speed and conventional rolling stock can be operated together, but at a speed penalty. Iannone et al. (see [9]) assess the impact of complex management decisions on planning and operations of inter-modal logistical terminals. Based on data from Italian terminals, they model both scheduling and loading/unloading processes of cargo on large marine vessels, finding that by adopting a different scheduling method a two digit percentage of monetary cost can be saved. Leonard et al. (see [14]) analyze the launch process of the NASA Space Launch System, focusing on the supply of the four main liquid launch commodities: hydrogen, oxygen, nitrogen, and helium. Their study confirms that, applying a new refilling strategy, the turnaround time could be reduced from over seven days to 48 hours. Schroer et al. (see [22]) model a number of terminals of the port of Rotterdam, Netherlands. They investigate options for the transport of cargo containers between these sub-systems, and consider several discussed configurations of the port's future extensions.

Transit: Ullrich et al. (see [24]) use a discrete-event simulation model of light-rail transit to examine the conditions under which the application of regular timetables reduces service delays resulting from inevitable small disturbances. They find that for regular timetables to have an impact, the light-rail system has to adhere to three conditions: resources like tracks and splits have to be shared, the traversal time variance has to be comparatively low, and resources cannot be redundant. Wales and Marinov (see [26]) describe delayed service issues in the British Tyne and Wear Metro light-rail system. They develop a simulation model to analyze the origins of these delays, and to assess a number of potential mitigation techniques. They find that by applying these techniques, the system's punctuality could be improved by 15% to 17% in relation to the current state.

A collection of further recent use cases, especially in the areas of automotive and manufacturing simulation, can be found in [4].

5 Conclusions

This paper presented an introduction to discrete-event modeling and simulation, in particular to the event-oriented worldview. It shared some fundamental modeling techniques, described the execution of simulation models, and introduced *Surfside*, a slim framework for event-oriented simulation. An overview of some current case studies showed the relevance of the paradigm for researchers and practitioners.

Discrete-event modeling and simulation is a very simple, easy to understand, but also very powerful approach. It is especially useful for modeling technical systems in the wider sense, including communications, disaster mitigation, health care, logistics, supply management, and transportation. With the help of slim frameworks a single researcher or practitioner can build, execute, and evaluate a small but meaningful discrete-event model.

For further, more detailed research a number of books and articles can be recommended: Banks et al. (see [5]) is a seminal work on discrete-event modeling and simulation. Fishman (see [7]) is an excellent alternative, especially for readers with an interest in operations research. Model verification and validation are the single most important tasks in any simulation project. Sargent (see [21]) describes a set of paradigms and techniques to ensure a model's high validity and credibility. Even 15 years after its release, Fujimoto (see [8]) is still the weapon of choice for those who want to understand the fundamentals of parallel and distributed simulation. For modelers who desire to take a step beyond discrete event mechanics, Macal and North's introduction to agent-based modeling and simulation (see [17]) can be recommended.

6 Acknowledgments

This material is based in part upon work supported by the National Science Foundation under Grant Nos. I/UCRC IIP-1338922, AIR IIP-1237818, III-Large IIS-1213026, MRI CNS-1532061, MRI CNS-1429345, MRI CNS-0821345, MRI CNS-1126619, MRI CNS-0959985, RAPID CNS-1507611, and U.S. DOT Grant ARI73.

References

[1] Abbott D, Marinov MV. An event based simulation model to evaluate the design of a rail interchange yard, which provides service to high speed and conventional railways. In: Simulation Modelling Practice and Theory, Vol. 52, 2015, pp. 15-39.

[2] Allen RW, Taaffe KM, Ritchie G. Surgery Rescheduling Using Discrete Event Simulation: A Case Study. In: Proceedings of the 2014 Winter Simulation Conference, 2014, pp. 1365-1376.

[3] Artuso M, Lehrmann Christiansen H. Modeling and Event-Driven Simulation of Coordinated Multi-Point Joint Transmission in LTE-Advanced with Constrained Backhaul. In: Proceedings of the 2014 Winter Simulation Conference, 2014, pp. 3131-3142.

[4] Bangsow S. (Ed.): Use Cases of Discrete Event Simulation. Springer, 2012.

[5] Banks J, Carson JS, Nelson BL, Nicol DM. Discrete-Event System Simulation, 5th edition. Pearson, 2010.

[6] Cetinkaya EK, Broyles D, Dandekar A, Srinivasan S, Sterbenz JPG. Modeling communication network challenges for Future Internet resilience, survivability, and disruption tolerance: a simulation-based approach. In: Telecommunication Systems, Vol. 52, 2013, pp. 751-766.

[7] Fishman GS. Discrete-Event Simulation: Modeling, Programming, and Analysis. Springer, 2001.

[8] Fujimoto R. Parallel and Distributed Simulation Systems. John Wiley & Son, 2000.

[9] Iannone R, Miranda S, Prisco L, Riemma S, Sarno D. Proposal for a flexible discrete event simulation model for assessing the daily operation decisions in a Ro-Ro terminal. In: Simulation Modelling Practice and Theory, Vol. 61, 2015, pp. 28-46.

[10] Knuth D. The Art of Computer Programming, Volume 2: Seminumerical Algorithms, 3rd Edition. Addison-Wesley, 1997.

[11] Kuchel D. Analyzing Noncombatant Evacuation Operations Using Discrete Event Simulation. In: Proceedings of 2013 Winter Simulation Conference, 2013, pp. 2751-2761.

[12] Kuckertz P, Randerath H. A model for airline personnel schedule simulation. In: Proceedings of ASIM-Workshop STS/GMMS 2014, ARGESIM Report 42, ASIM Mitteilung AM 149, ARGESIM/ASIM Pub., TU Vienna/Austria, Februar 2014, pp. 171-180.

[13] Kuckertz P, Ullrich O, Randerath B. A simulation based approach on robust airline job pairing. In: Simulation Notes Europe (SNE), Volume 22, Number 2, 2012, pp. 77-82.

[14] Leonard D, Parsons J, Cates G. Using discrete event simulation to model fluid commodity use by the space launch system. In: Proceedings of the 2014 Winter Simulation Conference, 2014, pp. 2954-2965.

[15] Lueckerath D. Ein Simulationsmodell fuer Oeffentlichen Personennahverkehr mit regelbasiertem Verkehrsmanagement. Dissertation, Univ. Koeln, 2016.

[16] Lueckemeyer G. A Traffic Simulation System Increasing the Efficiency of Schedule Design for Public Transport Systems Based on Scarce Data. Dissertation, Univ. Koeln, 2007.

[17] Macal CM, North MJ. Tutorial on agent-based modeling and simulation. In: Journal of Simulation, Vol. 4, 2010, pp. 151-162.

[18] Mayorga ME, Wheeler SB, Reifsnider OS, Kohler RE. A Discrete Event Simulation Model to Estimate Population Level Health and Economic Impacts of Smoking Cessation Interventions. In: Proceedings of the 2014 Winter Simulation Conference, 2014, pp. 1257-1268.

[19] Nance RE. A history of discrete event simulation programming languages. In: Proceedings of the 2nd ACM SIGPLAN conference on History of programming languages, 1993, pp. 149-175.

[20] Pooya P, Ivy J, Mazur L, Deschesne K, Mosaly P, Tracton G. Assessing the Reliability of the Radiation Therapy Care Delivery Process Using Discrete Event Simulation. In: Proceedings of the 2014 Winter Simulation Conference, 2014, pp. 1233-1244.

[21] Sargent RG. Verification and validation of simulation models. In: Journal of Simulation, Vol. 7, No. 1., 2013, pp. 12-24.

[22] Schroer HJL, Corman F, Duinkerken MB, Negenborn RR, Lodewijks G. Evaluation of inter terminal transport configurations at Rotterdam Maasvlakte using discrete event simulation. In: Proceedings of the 2014 Winter Simulation Conference, 2014, pp. 1771-1782.

[23] Suk Na H, Banerjee A. An Agent-Based Discrete Event Simulation Approach for Modeling Large-Scale Disaster Evacuation Networks. In: Proceedings of the 2014 Winter Simulation Conference, 2014, pp. 1516-1526.

[24] Ullrich O, Lueckerath D, Speckenmeyer E. Do regular time tables help to reduce delays in tram networks? - It depends! In: Public Transport, Volume 8, Issue 1, 2016, pp. 39-56.

[25] Ullrich O, Azzouzi C, Brandt B, Lueckerath D, Rishe N. Modeling Garage Parking. In: Simulation Notes Europe (SNE), Volume 26, Number 1, pp. 1-8.

[26] Wales J, Marinov M. Analysis of delays and delay mitigation on a metropolitan rail network using event based simulation. In: Simulation Modelling Practice and Theory, Vol. 52, 2015, pp. 52-77.

Parallel Multi-agent Smart Grid Simulation

Christian Kuschel*, Ulrich Rüde

Friedrich-Alexander-Universität Erlangen-Nürnberg, Lehrstuhl für Systemsimulation (Informatik 10), Cauerstr. 11, 91058 Erlangen, Germany; *christian.kuschel@fau.de

SNE 27(1), 2017, 17-23, DOI: 10.11128/sne.27.tn.10363
Received: February 10, 2017, (Selected ASIM STS 2016 Post-conf. Publ.), Accepted: February 25, 2017
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. The approach in this paper renders it possible to simulate large-scale smart grids by efficient parallel computations. This permits a detailed analysis of the consumption behaviours, efficiency and impact of green energies, and self-sustainability of a smart grid. The smart grid is modelled as a multi-agent system. Each agent represents a building which is optimally controlled. That is, an agents meets its prescribed energy demand by trading energy or applying devices, e.g. solar panels and fuel cells, minimising its costs. A cooperative bargaining game is devised in which the agents participate to obtain a global optimal solution. In this paper, this inherently serial bargaining game is parallelised. The parallelisation is necessary to be able to deal with the large amount of data and computations which need to be performed. In the experiments the validity of the presented approach is shown and as a proof of concept a large smart grid of over 40 million agents is simulated.

Introduction

With technologies for integrating energy generation and storage in residential buildings, the notion of smart grid was derived [1, 2]. Smart grids have been in the focus of research because of various aspects. Among others, the energy generation and storage devices themselves, e.g. combined heat and power generation and energy storages, and the coordination of them pose many challenges [3, 4, 5, 6, 7]. The focus of this paper, however, is on the interactions of participants of the smart grid.

The smart grid is modelled by a multi-agent system. Each agent optimally controls a residential buildings and is able to generate and store energy with the appropriate devices [8, 9]. Based on the models of [10, 11], the agents are able to communicate with each other to participate in a cooperative bargaining game which possesses a unique Nash equilibrium. In addition, the pricing scheme from [13] is applied to obtain fair energy prices during bargaining.

The efforts of this paper go beyond the previously mentioned approaches of simulating smart grids: The smart grid model used in this paper extends the model [10] by incorporating a more detailed agent model which was published in [11]. In both publications a serial, weak coupling approach for solving the bargaining game is realised. The goal of this paper is enable simulations of realistically sized, large smart grids. Therefore the serial approach is parallelised for a distributed memory architecture, because a single computer can no longer deal with the intended complex and memory consuming simulation.

With this parallelisation, detailed information about the optimal, efficient usage of energy can be obtained. In particular, the efficient usage of green energies and the self-sustainability of the smart grid can be examined. To that end, a smart grid of more than 40 million agents, which is of equivalent size of Germany [12], is simulated in parallel. To the knowledge of the authors, this is the largest smart grid simulation performed until today.

In Section 1, a brief overview about the applied model and the serial algorithmic approach is given. The parallelisation of the cooperative bargaining game is presented in Section 2. Finally, the experimental validation and performance results are presented in Section 3. The test results validate the approach and show the parallel performance in strong scaling experiments.

1 Smart Grid Model

The smart grid model and the solution approach are essentially based on [10, 11, 13]. In the following a brief summary of the method published in [11] is given. Consider a smart grid as depicted in Figure 1.

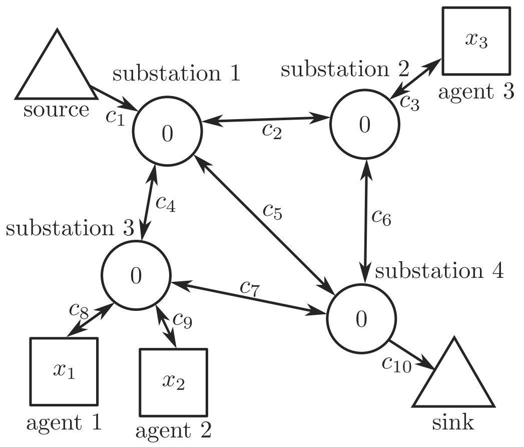


Figure 1: Schematic illustration of a smart grid with line capacities. The differently shaped nodes indicate different roles. The values within nodes specify the required net load.

In this paper the line capacities c_i of Figure 1 are assumed to be infinite, so that the grid does not pose any restrictions and can be neglected.

A smart grid consists of agents and a grid operator which are capable of communicating with each other. The grid operator uses power plants to generate energy and substations to distribute the energy. Since the line capacities are infinite, the substations can be neglected.

Buildings are optimally controlled by agents by a solving optimisation problem. The optimisation problem takes devices, like e.g. central heating, fuel cells, refrigerators, into account, and enforces that a prescribed energy demand is met. This is incorporated in constraints of the optimisation problem. The constraints define a non-empty, convex feasible set Ω_n for each agent n. The details about the feasible set are not relevant for the considerations below, but can be found in [11]. It is sufficient to assume that the set is non-empty (to ensure feasibility) and that the choice of the optimal solution is restricted in one way or the other (excluding a trivial solution).

The agents' optimisation problems are solved to minimise their respective costs. Since the goal is to simulate millions of agents, it is assumed that the agents form a market which determines the price with respect to demand and supply. Therefore, the agents can adapt their demand to the prices and influence the prices by changing the demand. This adaption is modelled by a cooperative bargaining game to minimise the agents' costs which is presented below.

The total incurring costs $\mathscr{C}$ of the budget balanced grid operator to supply the agents is defined by

$$\mathscr{C} := \left(\sum_{n=1}^{N} x_n\right)^2,$$

where N is the number of agents and x_n is the net load of agent n. The net load subsumes an agent's demand and supply in one variable. If $x_n > 0$ it represents a demand, if $x_n < 0$ it represents a supply. The cost function $\mathscr{C}$ can be understood as the squared deviation of a self-sustaining smart grid. If $\mathscr{C} \neq 0$ the grid operator must intervene and costs incur which must be covered by the agents.

Based on the fair pricing scheme suggested by [13], the total costs are split among the agents proportionally to their contribution to the total costs. This yields the individual incurring costs $\mathscr{C}_n$ of agent n with respect to all other agents' loads x_{-n}:

$$\begin{aligned} \mathscr{C}_n(x_n; x_{-n}) &= \overbrace{(x_n + x_{-n})}^{\text{price}} \cdot \overbrace{x_n}^{\text{net load}}, \\ &\quad \text{with } x_{-n} := \sum_{\substack{j=1 \\ j\neq n}}^{N} x_j. \end{aligned} \qquad (1)$$

So each agent minimises (1), the local objective function, in an optimisation problem. In particular, $\mathscr{C} = \sum_{n=1}^{N} \mathscr{C}_n(x_n; x_{-n})$.

In order to obtain a global optimum, a bargaining game is devised:

- Players: All agents in the smart grid.
- Strategies: Player n computes its best response $x_n = \arg\min \mathscr{C}_n(x_n, x_{-n})$ s.t. $x_n \in \Omega_n$.
- Costs: $\mathscr{C}_n$ for agent n.

For the considered game it was shown in [10] that a unique Nash equilibrium exists. Therefore, the locally computed optima by the agents lead to a global optimum. This Nash equilibrium is characterised by each player obtaining its optimum. Moreover, if one player deviates from the optimum solution, the costs increase for that player.

Since the costs of one agent are dependent on the net loads of all other agents, as indicated in (1), the game is played in rounds. In each round, all agents adjust their played net load to their respective current best responses after another. This is done until no adjustments are carried out by all agents. This procedure of the game is summarised in Algorithm 1. In the subsequent section, this algorithm is parallelised.

```
$k \leftarrow 0$
Set $x^k$ to initial total grid load
repeat
    $\delta \leftarrow 0$
    $y^k \leftarrow x^k$
    for all agent $n$ in the smart grid do
        $x_{-n}^k \leftarrow y^k - x_n^k$
        $x_n^{k+1} \leftarrow \arg\min \mathscr{C}_n(x_n^{k+1}; x_{-n}^k)$ s.t. $x_n^{k+1} \in \Omega_n$
        $y^k \leftarrow x_n^{k+1} + x_{-n}^k$
        $\delta \leftarrow \delta + |x_n^k - x_n^{k+1}|/N$
    end for
    $x^{k+1} \leftarrow y^k$
    $k \leftarrow k+1$
until $\delta < \varepsilon$
```

Algorithm 1: Serial bargaining algorithm.

2 Parallel Bargaining Game

In order to parallelise Algorithm 1, the best response computation is distributed on processes which run in parallel. In general, more agents than processes are used, therefore one process is assigned multiple agents for computation. Thus each process computes the best responses of its agents serially as in Algorithm 1. This can be interpreted as computing a single best response for each process, independent of the actual number of agents associated with the process. So instead of referring to the agents' best responses, below the processes' best responses are considered.

An essential property of the game in Algorithm 1 is the successive best response computation. Evidently, when processes compute the best responses in parallel as described above, this property is violated. Consequently, for the parallel approach a synchronisation scheme needs to devised which is applied after the processes' best response computation to compensate the lack of the successive best response computations.

Consider again Algorithm 1. Since the best response computation is the solution to the agents' optimisation problems, this entails that the process' costs never increase. This non-increase property is expressed as

$$\mathscr{C}_n(x_n^{k+1}; x_{-n}^k) \leq \mathscr{C}_n(x_n^k; x_{-n}^k) \tag{2}$$

and must hold for every iteration k. This also implies that $|x_n^{k+1}| \leq |x_n^k|$ for all $n = 1, \ldots, N$. In fact, the objective value monotonically decreases until a global optimum has been found [10, 14].

The monotonic decrease in (2) holds for each process individually. However it must also hold for the total incurring costs in the smart grid (as in the serial case), therefore

$$\mathscr{C}^{k+1} \leq \mathscr{C}^k := \sum_{n=1}^{N} \mathscr{C}_n(x_n^k; x_{-n}^k). \tag{3}$$

In the parallel approach, a monotonic decrease in the total costs is obtained by computing the best response with respect to an auxiliary term $\bar{x}^k$, which represents the total net load in the grid. Let

$$\bar{x}_{-n}^k := \bar{x}^k - x_n^k.$$

So instead of minimising $\mathscr{C}_n(x_n^{k+1}; x_{-n}^k)$, $\mathscr{C}_n(x_n^{k+1}; \bar{x}_{-n}^k)$ is minimised.

The auxiliary term is defined as

$$\bar{x}^{k+1} = \frac{N-1}{N^2} \cdot \sum_{n=1}^{N} \left(x_n^{k+1} + \bar{x}_{-n}^k \right) + \frac{1}{N} \cdot \sum_{n=1}^{N} x_n^{k+1}. \tag{4}$$

It can be shown that (3) holds for this choice of $\bar{x}^k$. To show that $(\bar{x}^k)^2 \geq (\bar{x}^{k+1})^2$, the reduction γ is introduced to yield $(\bar{x}^k - \gamma)^2 = (\bar{x}^{k+1})^2$. Applying the definition of $\bar{x}^{k+1}$ from (4) reads

$$\left(\bar{x}^k - \gamma\right)^2 = \left(\bar{x}^k - \frac{1}{N}\bar{x}^k \right. \\ \left. + \frac{N-1}{N^2} \cdot \sum_{n=1}^{N} \left(\frac{2 \cdot N - 1}{N-1} \cdot x_n^{k+1} - x_n^k \right) \right)^2.$$

Obviously,

$$\gamma = \frac{1}{N}\bar{x}^k - \frac{N-1}{N^2} \cdot \sum_{n=1}^{N} \left(\frac{2 \cdot N - 1}{N-1} \cdot x_n^{k+1} - x_n^k \right) \tag{5}$$

must be within $0 \leq \gamma \leq 2 \cdot \bar{x}^k$ for $\bar{x}^k \geq 0$ or $0 \leq -\gamma \leq 2 \cdot \bar{x}^k$ for $\bar{x}^k \leq 0$ to yield a decrease in the objective value.

In the following the case of $\bar{x}^k \geq 0$ is elaborated. The case $\bar{x}^k \leq 0$ can be derived analogously. To show that $\gamma \geq 0$, (5) is rewritten as

$$\frac{1}{N} \cdot \left(\bar{x}^k - \sum_{n=1}^{N} x_n^{k+1} \right) - \frac{N-1}{N^2} \cdot \sum_{n=1}^{N} \left(x_n^{k+1} - x_n^k \right).$$

Both terms in parenthesis represent the change in the objective value from iteration k to $k+1$. The term in the first parenthesis is the change with respect to the auxiliary term $\bar{x}^k$ whereas the term in the second is the change of the best responses $\sum_{n=1}^{N} x_n^k$. Both expressions in parenthesis are negative, but due to the averaging of $\bar{x}^k$ the term in the first parenthesis is more inertial to change than the second term. Thus the absolute value of the term in second parenthesis is larger so that $\gamma \geq 0$ holds.

(5) is inserted to $\gamma \leq 2 \cdot \bar{x}^k$ to show this inequality also holds:

$$(2 \cdot N - 1) \cdot \bar{x}^k \geq \frac{2 \cdot N - 1}{N} \sum_{n=1}^{N} x_n^{k+1} - \frac{N-1}{N} \sum_{n=1}^{N} x_n^k.$$

Recall that $|x_n^{k+1}| \leq |x_n^k|$ from (2) is implied, so

$$(2 \cdot N - 1) \cdot \bar{x}^k \geq \left(\frac{2 \cdot N - 1}{N} - \frac{N-1}{N} \right) \sum_{n=1}^{N} x_n^k = \sum_{n=1}^{N} x_n^k$$

is obtained. In the course of iteration, $\bar{x}^k$ approximates $\sum_{n=1}^{N} x_n^k$. Consequently, both terms are within the same order and the estimate

$$\bar{x}^k \leq \sum_{n=1}^{N} x_n^k \leq 2\bar{x}^k < (2N-1) \cdot \bar{x}^k \Rightarrow \left| \bar{x}^k - \sum_{n=1}^{N} x_n^k \right| \leq |\bar{x}^k|$$

holds. $|\bar{x}^k - \sum_{n=1}^{N} x_n^k|$ is generally small and decreases with increasing number of iterations k, since the x_n^k are computed as solution to optimisation problems and used to construct $\bar{x}^k$. Therefore (4) yields a monotonic decrease.

With the monotonic decrease shown in each iteration, it yet needs to be validated that the sequence of $\bar{x}^k$ converges to the optimum. This is done by showing convergence of $\bar{x}^k \to \sum_{n=1}^{N} x_n^k$ for $k \to \infty$. By expressing $\bar{x}^k$ with respect to the best responses x_n^{k+1} and $\bar{x}^k$ the following recurrence equation for a given initial value $\bar{x}^0 = \sum_{n=1}^{N} x_n^0$ is obtained:

$$\begin{aligned}\bar{x}^{k+1} = &\left(\frac{N-1}{N} \right)^{k+1} \cdot \bar{x}^0 + \frac{2 \cdot N - 1}{N^2} \cdot \sum_{n=1}^{N} x_n^{k+1} \\ &+ \frac{N-1}{N^2} \cdot \sum_{m=1}^{k} \left(\frac{N-1}{N} \right)^{k-m+1} \cdot \sum_{n=1}^{N} x_n^m. \qquad (6)\end{aligned}$$

Since the sequence $\bar{x}^k$ is monotonically decreasing, i.e. converging to a finite limit, the subsequence x_n^m is also convergent. So the sum in (6) containing x_n^m can be bound by a geometric series (since $|1/N| < 1$) by replacing x_n^m with x_n^k. The geometric series can be rewritten in a closed form. Using $x^* = \sum_{n=1}^{N} x_n^k$ for $k \to \infty$, from (6)

$$\lim_{k \to \infty} \bar{x}^{k+1} = 0 + \left(\left(\frac{N-1}{N} \right)^2 + \frac{N-1}{N^2} + \frac{1}{N} \right) \cdot x^* = x^*$$

is obtained. Consequently, using an auxiliary term as defined in (4) implies convergence to the optimum. Thus, the serial bargaining game can be parallelised by this method.

Concluding this section, the parallel bargaining is summarised in Algorithm 2. The new variable P in line 15 denotes the total number of agents in the entire smart grid. This number is required to check the termination criterion in line 17.

```
 1: Set $X$ to the initial net load
 2: $X \leftarrow$ allReduce$(X, +)$
 3: $k \leftarrow 0$
 4: **repeat**
 5:     $Y, \delta \leftarrow 0$
 6:     **for all** agent $j$ of process $n$ **do**
 7:         $X \leftarrow X - x_j^k$
 8:         $x_j^{k+1} \leftarrow \arg\min \mathscr{C}_j(x_j^{k+1}; X)$ s.t. $x_j^{k+1} \in \Omega_j$
 9:         $X \leftarrow X + x_j^{k+1}$
10:         $Y \leftarrow Y + x_j^{k+1}$
11:         $\delta \leftarrow \delta + |x_j^k - x_j^{k+1}|$
12:     **end for**
13:     $X \leftarrow$ allReduce$(X, +)$
14:     $Y \leftarrow$ allReduce$(Y, +)$
15:     $\delta \leftarrow$ allReduce$(\delta, +)/P$
16:     $X \leftarrow (N-1) \cdot X/N^2 + Y/N$
17:     $k \leftarrow k + 1$
18: **until** $\delta < \varepsilon$ and $|X - Y|/N < \varepsilon$
```

Algorithm 2: Parallel bargaining algorithm.

Each process performs the computations for its assigned agents independent of other processes on a distributed memory architecture. The "allreduce"-operation is the only operation which requires interprocess communication. It acts as barrier and all processes wait for each other at this point. When it is called, each process collects the passed value from the other processes and sums up the values, respectively. The return value of this function is the same for all processes.

Although in Algorithm 2 it is described that three communication steps in lines 14 to 16 are required, all reductions can be fused in only one reduction by writing all three variables in one buffer and perform one reduction on the buffer instead of one per variable.

3 Simulation Results

Experiments were conveyed to validate the approach of Section 1 and also analyse the parallel performance of the presented bargaining algorithm. The algorithms were implemented in C++ using the GNU linear programming kit (glpk) [15] as central solver for optimisation problems and Open MPI [16] for the parallelisation and interprocess communication.

In Section 3.1, the validity of the approach from Section 2 is shown. The parallel performance is analysed with respect to strong scaling in Section 3.2.

3.1 Validation of the approach

The validation was carried out for a reference smart grid with 10,000 agents. The smart grid is simulated for 100 time steps, which represents roughly one day in 15 minutes intervals. 200 bargaining iterations were performed in each case.

As shown in Figure 2, all plots converge to the reference result, which was computed serially by the approach of [11]. So the approach as described in Algorithm 2 allows to compute the optimum in parallel. It can be observed that, although convergent, the more processes are used, the slower the convergence. This can be traced back to the coefficients dependent on N in (6) which tend to 1 for large N. Therefore the more processes are used, the slower the convergence rate.

Although the plot shows that the number of required iterations to reach the optimum increases with increasing number of processes, the total runtime does not increase. These details of the performance are presented in the next section.

3.2 Parallel performance

The efficiency of the implementation is considered in the following. Therefore the strong scaling behaviour of the parallel bargaining approach is presented. The termination criterion was set to a maximal average change of net load per agent per time step of 10^{-2} kWh. Furthermore, the agents are uniformly distributed on the processes in a round robin fashion.

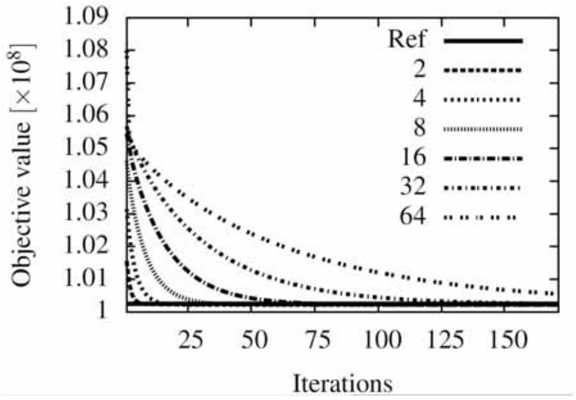


Figure 2: Convergence of parallel bargaining for 2 to 64 processes, and the reference solution.

The strong scaling test scenario is the same as in Section 3.1. It was computed on the local compute cluster of the department of computer science 10 of Friedrich-Alexander-Universität Erlangen-Nürnberg [17]. It has eight compute nodes each of which consists of four Intel Xeon E7-4830 (eight cores each) at 2.4 GHz and 256 GB RAM per node. The serially computed scenario took 42.5 minutes to finish.

As already mentioned, in Figure 2 it can be observed that the more processes are used the more iterations are required to attain the optimum. This, however, does not entail a longer total runtime, as can be seen in Table 1. In fact, since the optimisation problems of the agents are uniformly distributed, the time per iteration approximately halves when using twice as much processors. The number of iterations hardly increases with increasing number of processes. Consequently, the parallelisation has a greater impact than the number of increased iterations, therefore the total runtime decreases.

In Table 1, it is stated that there is close to perfect speed-up up to eight processes. Beyond that the speed-up of the computations increases less. The reason for that are the increased communication in addition to the fluctuating times to solution of an agent's optimisation problem. For the considered scenario the latter times are between 2 s and 0.05 s with a median of 0.07 s. Obviously, most of the agents' optimisation problems are solved quickly and few slowly. If the optimisation problems are distributed among the processes in a way that the average time to solve all optimisation problems per process is approximately equal, a good speed-up can be expected.

For the considered scenario more than 1,250 agents

# Processes	1	2	4	8	16	32
# iterations	3	4	5	5	6	6
it. time $[s]$	798	349	181	81	37	21
Speed-up	1	1.7	2.8	6.0	10.9	20.1

Table 1: Strong scaling results for 10,000 agents, showing the number of iterations, the averaged iteration time and the speed-up with respect to the serial computation.

per process yielded excellent scaling. Fewer agents caused some processes to wait for other, slower processes reducing the effects of parallelisation. This effect is expected to be alleviated by using dynamic instead of the static load balancing used in the experiments. Nevertheless the parallel efficiency does not drop below 60% in the experiments, which indicates good scalability.

In addition to the strong scaling computations, one large simulation as proof of concept for the realisation of a nation-sized smart grid was carried out. Algorithm 2 was executed on the Emmy cluster of Regionales Rechenzentrum Erlangen [18] simulating 40 million agents on 1280 processes. This was the largest simulation with respect to the physical limitations of the compute nodes and it completed within 16 hours. This scenario corresponds to a smart grid in the order of Germany, which is the country with the most private households in the EU [19]. To the knowledge of the authors, this is the largest simulated smart grid until today.

This carried out experiment merely hints at the potential of this approach. Being able to simulate smart grids representing countries, analysis of efficiency and sustainability can be performed. Especially if investments pay off can be analysed. From the simulation aspect, the weak scaling has yet to be examined. This is necessary to determine the efficiency of this approach with increasing number of agents on an increasing number of processes. This is analysed in the current work in progress.

4 Conclusion and Future Work

In this paper, a cooperative bargaining game was successfully parallelised. Since the serial algorithm relies on successive best response computation, it is necessary to introduce an auxiliary term to ensure convergence of the approach.

Almost perfect speed-up is reached in the strong scaling scenarios, in which 10,000 agents are considered, if the average time to solution per process is almost equal. This was commonly the case for more than 1,250 agents per process in the experiments.

In a next step the parallel efficiency needs to be evaluated for weak scaling scenarios. When simulating large smart grids, the presented algorithm is required to scale well with the problem size. In conclusion, the largest run optimised more than 40 million agents, a problem size equivalent to a country. Being able to simulate such a large grid, estimates on a realistic scale about self-sufficiency, demand and supply and efficiency of green energies can be fleshed out.

References

[1] Biswas MdM et al. *Towards Implementation of Smart Grid: An Updated Review on Electrical Energy Storage Systems*. In: Smart Grid and Renewable Energy 4.4 (2013), pp. 122–132.

[2] Monti A, Ponci F. *Power Grids of the Future: Why Smart Means Complex*. In: Complexity in Engineering, 2010. COMPENG '10. 2010, pp. 7–11.

[3] Houwing M, Negenborn R, De Schutter B. *Demand response with micro-CHP systems*. In: Proceedings of the IEEE 99.1 (Jan. 2011), pp. 200–213.

[4] Arif M, Oo A, Ali A. *Role of Energy Storage on Distribution Transformer Loading in Low Voltage Distribution Network*. In: Smart Grid and Renewable Energy 4.4 (2013), pp. 237–251.

[5] Atzeni I et al. *Demand-Side Management via Distributed Energy Generation and Storage Optimization*. In: Smart Grid, IEEE Transactions on 4.2 (2013), pp. 866–876.

[6] Giuntoli M, Poli D. *Optimized Thermal and Electrical Scheduling of a Large Scale Virtual Power Plant in the Presence of Energy Storages*. In: IEEE Transactions on Smart Grid 4.2 (2013), pp. 942–955.

[7] van de Ven P et al. *Optimal Control of End-User Energy Storage*. In: IEEE Transactions on Smart Grid 4.2 (2013), pp. 789–797.

[8] Karl J. *Dezentrale Energiesysteme: Neue Technologien im liberalisierten Energiemarkt*. Oldenbourg, 2006.

[9] Ekanayake J et al. *Smart Grid: Technology and Applications*. Wiley, 2012.

[10] Mohsenian-Rad AH et al. *Autonomous Demand-Side Management Based on Game-Theoretic Energy Consumption Scheduling for the Future Smart Grid*. In: IEEE Transactions on Smart Grid 1.3 (2010), pp. 320–331.

[11] Kuschel C, Köstler H, Rüde U. *Multi-Energy Simulation of a Smart Grid with Optimal Local Demand and Supply Management*. In: Smart Grid and Renewable Energy 6 (2015), pp. 303–315.

[12] Bundesamt für Statistik. *Staat & Gesellschaft – Haushalte & Familien – Statistisches Bundesamt (Destatis)*. www.destatis.de/DE/ZahlenFakten/GesellschaftStaat/Bevoelkerung/HaushalteFamilien/HaushalteFamilien.html. Feb. 2016.

[13] Baharlouei Z et al. *Achieving Optimality and Fairness in Autonomous Demand Response: Benchmarks and Billing Mechanisms*. In: IEEE Transactions on Smart Grid 4.2 (2013), pp. 968–975.

[14] Berninghaus S, Ehrhart KM, Güth W. *Strategische Spiele*. Springer-Lehrbuch. Springer-Verlag Berlin Heidelberg, 2006.

[15] GLPK. *GNU Linear Programming Kit*. www.gnu.org/software/glpk. July 2015.

[16] The Open MPI Project. *Open MPI: Open Source High Performance Computing*. www.open-mpi.de. Mar. 2016.

[17] Computer Science 10, Friedrich-Alexander-University Erlangen-Nuremberg. *HPC Cluster*. www10.cs.fau.de/en/research/hpc-cluster. Feb. 2016.

[18] Regionales Rechenzentrum Erlangen. *RRZE – Emmy Cluster*. www.rrze.fau.de/dienste/arbeiten-rechnen/hpc/systeme/emmy-cluster.shtml. Feb. 2016.

[19] Bundesamt für Statistik. *EU – Privathaushalte in den Mitgliedsstaaten 2014*. de.statista.com/statistik/daten/studie/74720/umfrage/anzahl-der-privathaushalte-in-den-laendern-der-eu. Mar. 2016.

EUROSIM 2019

9th EUROSIM Congress on Modelling and Simulation

La Rioja, Logroño, Spain, July 2019

EUROSIM Congresses are the most important modelling and simulation events in Europe. For EUROSIM 2019, we are soliciting original submissions describing novel research and developments in the following (and related) areas of interest: Continuous, discrete (event) and hybrid modelling, simulation, identification and optimization approaches. Two basic contribution motivations are expected: M&S Methods and Technologies and M&S Applications. Contributions from both technical and non-technical areas are welcome.

Congress Topics The EUROSIM 2019 Congress will include invited talks, parallel, special and poster sessions, exhibition and versatile technical and social tours. The Congress topics of interest include, but are not limited to:

Intelligent Systems and Applications
Hybrid and Soft Computing
Data & Semantic Mining
Neural Networks, Fuzzy Systems & Evolutionary Computation
Image, Speech & Signal Processing
Systems Intelligence and Intelligence Systems
Autonomous Systems
Energy and Power Systems
Mining and Metal Industry
Forest Industry
Buildings and Construction
Communication Systems
Circuits, Sensors and Devices
Security Modelling and Simulation

Bioinformatics, Medicine, Pharmacy and Bioengineering
Water and Wastewater Treatment, Sludge Management and Biogas Production
Condition monitoring, Mechatronics and maintenance
Automotive applications
e-Science and e-Systems
Industry, Business, Management, Human Factors and Social Issues
Virtual Reality, Visualization, Computer Art and Games
Internet Modelling, Semantic Web and Ontologies
Computational Finance & Economics

Simulation Methodologies and Tools
Parallel and Distributed Architectures and Systems
Operations Research
Discrete Event Systems
Manufacturing and Workflows
Adaptive Dynamic Programming and Reinforcement Learning
Mobile/Ad hoc wireless networks, mobicast, sensor placement, target tracking
Control of Intelligent Systems
Robotics, Cybernetics, Control Engineering, & Manufacturing
Transport, Logistics, Harbour, Shipping and Marine Simulation

Congress Venue / Social Events The Congress will be held in the City of Logroño, Capital of La Rioja, Northern Spain. The main venue and the exhibition site is the University of La Rioja (UR), located on a modern campus in Logroño, capital of La Rioja, where 7500 students are registered. The UR is the only University in this small, quiet region in Northern Spain. La Rioja is where the Monasteries of San Millán de la Cogolla, cradle of the first words written in the Spanish language, are situated, sites included in UNESCO's World Heritage List in 1996. Of course, social events will reflect this heritage – and the famous wines in la Rioja.

CongressTeam: The Congress is organised by CAE CAE-SMSG, the Spanish simulation society, and Universidad de la Rioja.

Info: Emilio Jiménez, EUROSIM President, emilio.jimenez@unirioja.es
Juan Ignacio Latorre, juanignacio.latorre@unavarra.es

www.eurosim.info

Calibration Strategies for Agent-based Simulation Models with Variability

Claire Rippinger[1]*, Martin Bicher[2], Florian Miksch[3]

[1]Dexhelpp, Neustiftgasse 57-59, 1070 Vienna, Austria; *claire.rippinger@dexhelpp.at

[2]Institute for Analysis and Scientific Computing, TU Wien, Vienna Austria

[3]DWH Simulation Services, Vienna Austria

SNE 27(1), 2017, 25-31, DOI: 10.11128/sne.27.tn.10364
Received: November 5, 2016, Revised: January 20, 2017
Accepted: February 15, 2017
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna,
ISSN Print 2305-9974, Online 2306-0271 www.sne-journal.org

Abstract. During the development of an agent-based simulation model, the model often has to be calibrated, which means adjusting the parameters such that a reference system can be reproduced. A major problem in calibrating an agent-based simulation model is the variability of the results, due to random choices made by the agents. To reduce the variability, the numbers of agents has to be increased, which in return increases the computation time of the simulation. An attempted solution to this problem consists of increasing the numbers of agents gradually. This approach is tested with two different calibration algorithm: simulated annealing and evolutionary algorithm. Different updating schedules are applied on a test model and examined in terms of their running time and their performance. It is shown that a evolutionary algorithm with an increasing agent count manages to produce similar results as a standard calibration using only half the computation time. To conclude, the best performing calibration process is used to calibrate an existing agent-based model simulating a well known past influenza epidemic.

Introduction

Agent-based simulation is a relatively new modelling technique [1]. It has experienced increasing application in several fields since it offers many benefits over other modelling methods [2]. According to Bonabeau, the main advantages of agent-based models are their flexibility, their natural way to describe a system, and their ability to produce an emergent behavior [3]. Contrary to other modelling techniques, it does not try to dictate the general behaviour of the system. Instead, it consists of several independent entities, called agents, which are given certain properties, behaviour and rules to change this behaviour. These agents interact with each other and their environment during a simulation run and produce the overall outcome of the system. A typical application for this sort of modelling is the simulation of epidemics.

An important step in developing a model, agent-based or other, is the calibration. It consists of adjusting the different parameters used in the simulation such that the simulated results match a given outcome. When the model is capable of reproducing a reference system, it can be used to test the outcomes of alternatives strategies in this reference system or to make predictions by simulating the reference system in the future. The nature of agent-based simulation models induces different problems regarding the calibration process. Since the result of the simulation emerges from the interaction between the agents, the outcome is hard to estimate. Therefore, it is difficult to say in what way the different parameters affect the simulation outcome. Only by running the simulation, the effects may be observed and appropriate parameter changes can be made. If the model requires only a few parameters, these adjustments can be made manually. With an increasing number of parameters, calibration algorithms are needed.

Since a calibration problem consists of minimizing the distance between the simulated data and the reference system, it can be seen as an optimization problem and algorithms from this application area can be used. The nature of agent-based simulation models requires calibration algorithms which regard the simulation as a black box and only have informations on the outcome of the simulation and not on the internal processes and calculations. The evolutionary algorithm and simulated annealing are two algorithms meeting this criterion. They are presented in Section 1.

A common problem in performing a calibration is the long computation time required to produce the results. Section 1 proposes varying the amount of agents used in the simulation as a solution to this problem. Different configurations of this method are applied to a test model. The configuration with the best performance is then used to calibrate a more complex agent-based simulation model. Both models are described in Section 2.

1 Methods

In a calibration process, the agent-based simulation model acts as a function: given a specific parameter set, it produces the simulated data points. These are passed to an error function which calculates the distance between the simulated data points and the data points that should be matched by the model. Often, a weighted Euclidean distance is used. This allows the error function to put more emphasis on the characteristic elements of the data of the reference system. The aim of a successful calibration is to find a parameter set which minimizes this distance. Thus a calibration can be considered as an optimization problem and the respective algorithms can be used.

In this paper, two different optimization algorithm are applied for model calibration: simulated annealing [4] [5] and evolutionary algorithm [6][7]. Flowcharts of these algorithms can be seen in Figure 1 and Figure 2.

In simulated annealing, accepting a point with a larger error should allow the algorithm to escape local minima and converge to a global minimum. The acceptance probability depends on the temperature and is decreased during the calibration process. The cooling schedule applied, as well as an appropriate choice of the neighbourhood of a point, have a large influence on the convergence of the algorithm. The evolutionary algorithm uses several points simultaneously to determine the global minimum. By choosing different methods of selecting the points and combining them to form new candidates, evolutionary algorithms can be adapted to suit the needs of many calibration problems. However, finding the optimal configuration is often difficult.

Agent-based models often have long computation times caused by high agents numbers. This is an important issue during calibration, when the model is simulated hundreds of times. The runtime can be reduced with a lower number of agents. However, the agents behaviour usually depends on random decisions. Hence, the simulation results underlie a variability. For high agent numbers, the variability is naturally low due to the law of large numbers. Lower agent numbers lead to an unwanted higher variability of the results.

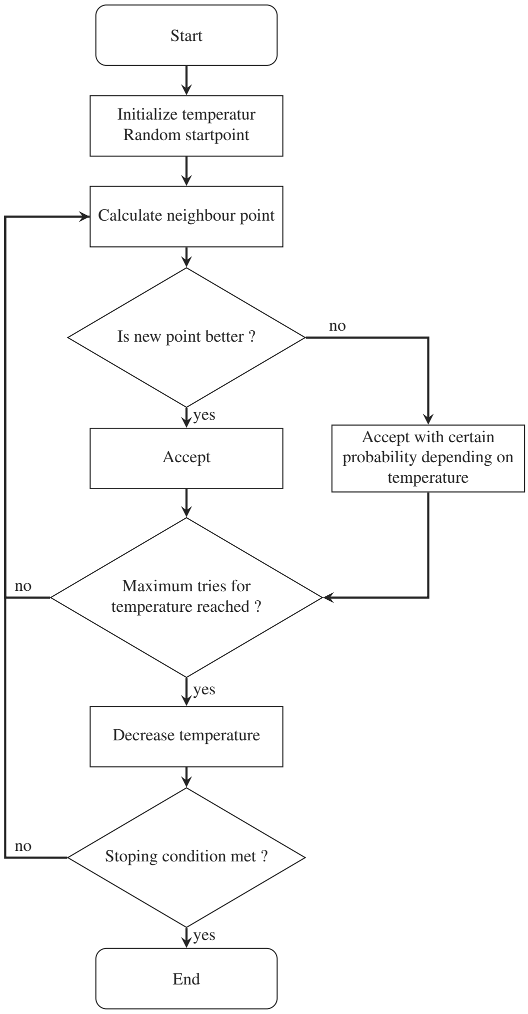


Figure 1: Flowchart of simulated annealing based on Kong et al. [8].

A possible solution to this problem consists of varying the number of agents throughout the calibration procedure. Agent-based models can be scaled by simulating them with reduced agent numbers. This does not affect its functionality but increases uncertainty of the results due to a required upscale to the actual problem. At the start, the simulation model is run with a small number of agents, allowing to test many parameter sets in a short time period. During the calibration the number of agents is gradually increased until the targeted agent count is reached.

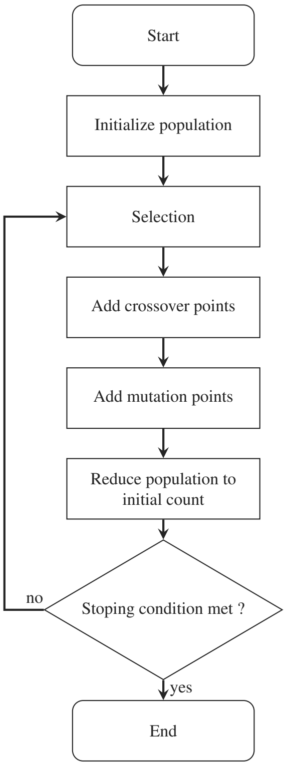


Figure 2: Flowchart of the evolutionary algorithm based on Kong et al. [8].

A calibration performing an increase of the agent count requires the following information:

- the starting amount of agents
- the targeted amount of agents
- the number of agent count updates
- the growth behaviour
- the number of simulation runs

After every update, the error of a newly considered parameter set is calculated with the new agent count. Normally, this error is only calculated once and then referred to in the further calibration process. However, this might cause a problem. If the error is calculated with a small agent count, it is possible that a low value has only been reached by accident and does not represent the usual outcome of the simulation performed with this parameter set. It is important to eliminate these false error values in time and not base every further search on the corresponding parameter sets. A simple solution would consist of recalculating the error of all the current parameter sets at every update of the agent count. This increases the number of simulation runs during a calibration, especially when an evolutionary algorithm is used. As a trade-off, the parameter sets are not re-evaluated directly at the next agent count update but after two agent count updates. This way, the chances are higher that the parameter set gets discarded by the algorithm before it needs to be recalculated. However, during the last update which increases the agent count to the targeted amount, every parameter set is re-evaluated to ensure that the best error has been calculated with the full agent count.

The increase of the agent count during the calibration has also an effect on the stopping conditions of the calibration algorithm. Normally, a calibration would terminate, if the error has reached a certain value. But, as mentioned above, if this error has been calculated with a small agent count, it might not be valid. Therefore the calibration is not allowed to terminate prematurely but has to perform the full amount of simulation runs.

2 Models

First, the calibration methods are tested on a simple SIR model which simulates the spreading of an infectious disease. In such a model, the agents represent people who can be in one of three different states: susceptible, infected or resistant. Every time, a susceptible person comes in contact with an person already infected the disease may be transmitted. After a certain amount of time, an infected person recovers from the disease and becomes resistant. This means, the person can not be infected a second time. In our model, there are two parameters that need to be calibrated: the probability p_i that a healthy person is infected when they comes in contact with an infected person and the probability p_r that an infected person is recovering. To create a reference system, the model is run with a known parameter set $p_i = 0.4$, $p_r = 0.05$. The output consists of the number of infected agents at a given time step. The goal of the calibration is to reproduce this curve. Previous tests have shown, that the variability of the results is negligibly small, if 10 000 agents are used, making this the target population.

Ultimately, the calibration algorithm is used to determine the parameters of a more complex agent-based model simulating an influenza epidemic. A known influenza epidemic in the year 2007 in the Austrian population is used as a reference system. The data that needs to be reproduced consists of eleven data points representing the number of newly infected people per week during the eleven week long influenza wave. For this model, the calibration needs to determine five parameters describing the contact rate between the agents, the probability of infection and development of mild or severe symptoms and the ratio of the population which is naturally immune. The reference system of both the simple SIR model and the influenza model can be seen in Figure 3.

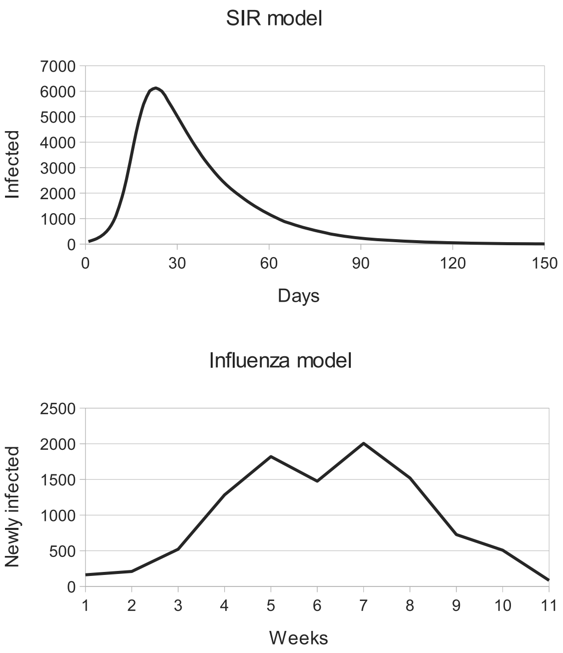


Figure 3: Reference systems of the SIR and the influenza model that need to be reproduced during the calibration.

3 Results

Both of the models described above, as well as the calibration algorithms have been implemented in Java and all the following calibrations have been calculated on a laptop running with Windows 8.1 using an Intel$^{(R)}$ Core$^{(TM)}$ i5-4200U processor and 8,00 GB RAM.

3.1 Results of calibration the SIR model

In order to compare the results of a calibration using an increasing amount of agents, the calibration has been performed with a constant agent count. The error is calculated using an Euclidean distance which puts a larger weight on the peak of the epidemic. In order to scale this error, the value of every data point is divided by the current agent count.

In this paper, 15 different update schedules are tested as shown in Table 1.

	Agents at Start	Growth	Updates
1	1000	no	0
2	3000	no	0
3	10 000	no	0
4	1000	geom.	2
5	1000	geom.	4
6	1000	geom.	8
7	3000	geom.	2
8	3000	geom.	4
9	3000	geom.	8
10	1000	linear	2
11	1000	linear	4
12	1000	linear	8
13	3000	linear	2
14	3000	linear	4
15	3000	linear	8

Table 1: Updating schedules.

The calibration is terminated after 1000 model runs. For each updating schedule the calibration is performed with three different calibration algorithm configurations providing good results in previous tests using a constant agent count. For each of these configurations, the calibration is run 10 times. For the simulated annealing, these configurations use a geometrical cooling schedule with the temperature being lowered every 10 loops by a factor 0.8, 0.85 resp. 0.9. In the evolutionary algorithm the population consists of 40 agents. A ranking based selection is applied [9]. During the crossover, 10 parameter sets are formed by calculating the mean of the two parent parameter sets and during mutation 8, 12 resp. 16 parameter sets are created by replacing one parameter with a random value.

Figure 4 shows the results of these calibrations. The bars represent the mean error of the 30 calibration results performed with the update schedule and the line represents the mean time to perform these calibrations. Note that the update schedule **3** represents a calibration performed with the targeted amount of agents throughout the whole process, making it the method applied

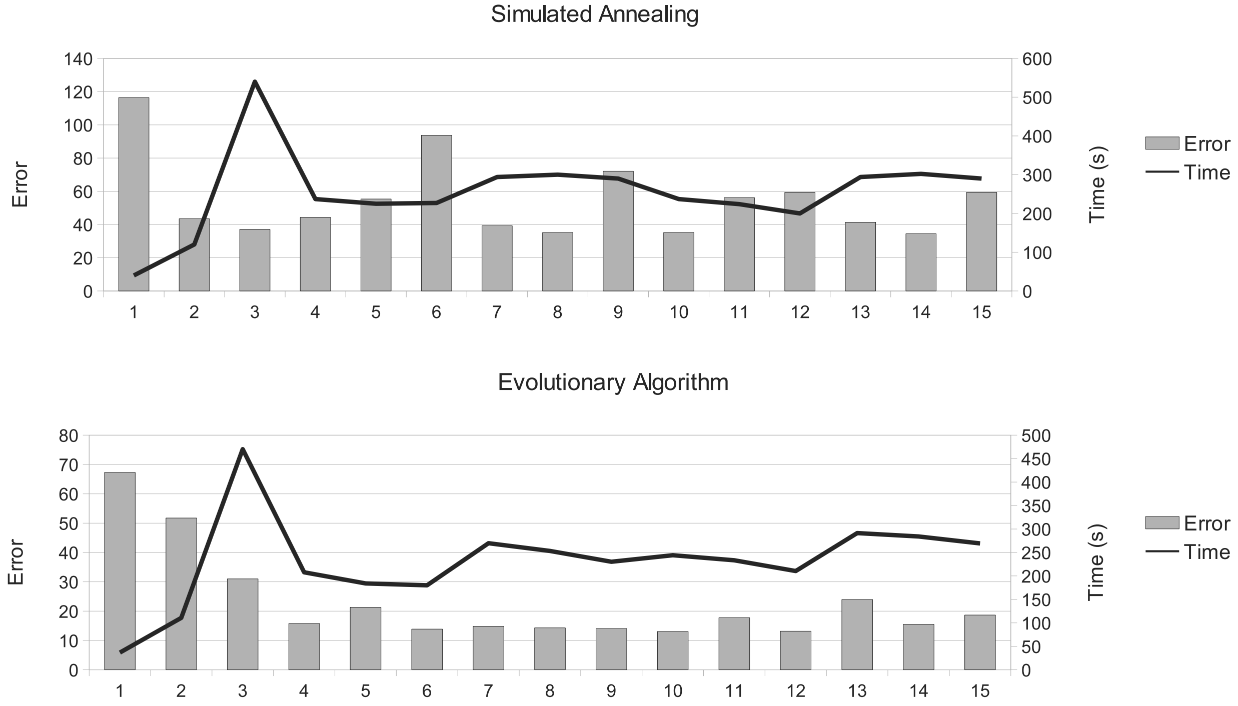


Figure 4: Error and running time of the calibration performed with simulated annealing and an evolutionary algorithm applying the different updating schedules listed in Table 1.

during a standard calibration. Update schedule **1** and schedule **2** represent calibrations using a constantly low agent count.

For the simulated annealing it can be seen, that only certain update schedules yield comparable results as a calibration performed with a constantly high amount of agents. Generally, it can be observed, that a higher number of agents at the start is preferable, as well as a smaller number of updates. The evolutionary algorithm produces overall better results than simulated annealing. The error obtained by updating the number of agents is even generally smaller than the one calculated by a standard calibration. There is no significant difference between the results of the different update schedules and no trend can be observed.

As expected, the computation time is much smaller for calibration updating the number of agents. In general, the running time is about half of the time required by a standard calibration represented by the update schedule **3**. Calculations using geometrical growth or a small number of starting agents require less time than those using linear growth and a higher amount of starting agents. Furthermore, the computation time decreases slightly with the number of updates performed. However it is expected that this decrease in running time is not an ongoing trend. At some point, the benefits of calculating with a lower agent count will be outweighed by the costs of re-evaluating the current population of parameter sets at every update. To verify this presentiment a new series of tests have been performed. Calibrations using geometrical growth and a starting agent count of 1000 are calculated using different numbers of updates, extending the calibrations **4**, **5** and **6** from Figure 4. The mean error and running time of these calibrations are shown in Figure 5. It can be seen, that the running time does increase with a larger number of updates without producing significantly better results.

It has been mentioned above that calibrations performed with simulated annealing provide worse results when the number of updates increases. Concerning this observation, further analysis of the error produced by these calibrations have revealed the following phenomenon: there are two different ways in which the er-

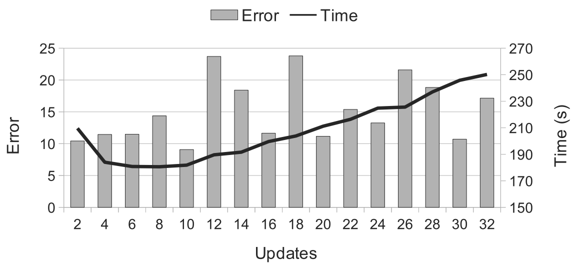


Figure 5: Error and running time for a calibration starting with 1000 agents and applying geometrical growth depending on the number of updates performed.

ror evolves. During some calibrations a relatively small error is already achieved using only a low agent count. The rest of the calibration process is then used for the fine tuning of the parameter set. However, if the error produced with a small amount of agents is not small, the current parameter set is not replaced with a better solution for the most part of the calibration. It is not until the simulation runs with the targeted amount of agents, that the calibration algorithm is effective and starts to lower the produced error. If the calibration process is performed using a high number of updates, the amount of simulation runs performed with the targeted agent count is too low to reach an acceptable error in time. Figure 6 shows the two different evolutions of the error during a calibration performed by simulated annealing using 4, resp. 8 updates. Each line represents one of the two typical behaviours of the error.

3.2 Results of calibration the influenza model

The findings of these tests are now used to calibrate the more complex influenza model. The error function used is similar to the one described at the beginning of this section. Since the data point at week six is most probably inaccurate, the weight posed on this point is very small. One simulation of the influenza model with 800 000 agents takes 300s on average. To improve the running time of the calibration, parallel computing on three kernels is applied. Since 1000 calibrations are performed, the total running time of a calibration without agent count updates would amount to approximately 28 hours. To further reduce this computation time, a calibration using 4 agent count updates is applied. The starting agent count consists of 50 000 agents which is increased geometrically to reach the targeted count of 800 000 agents. Due to the better results with the SIR model, the evolutionary algorithm is used for the calibration. The result can be seen in Figure 7. The running time of this calibration consisted of about 560 minutes, only a third of the estimated time required by a standard calibration.

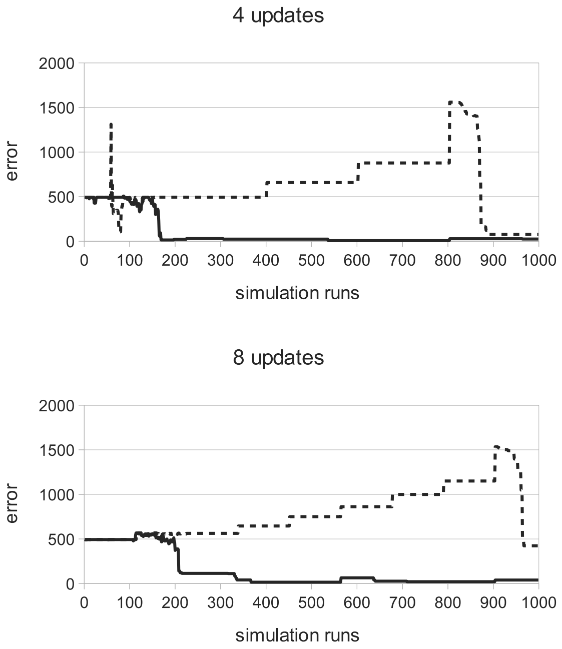


Figure 6: Two possible evolutions of the error calculated with simulated annealing during a calibration using geometrical growth, 1000 agents at the start and 4, resp. 8 agent count updates.

4 Discussion

This paper briefly describes the approach of an agent-based simulation model and the procedure of calibrating such a model. The variability of the results of an agent-based model complicate the task of calibration and the usual methods of reducing this variability lead to an increase in the running time of the calibration procedure. By gradually increasing the number of agents used in a simulation, this paper proposes a possible solution to this problem. This strategy is then tested on a simple agent-based simulation model. The performance looks very promising, but leaves a few open questions that require further research. For example, it might be possible to improve the performance of the simulated

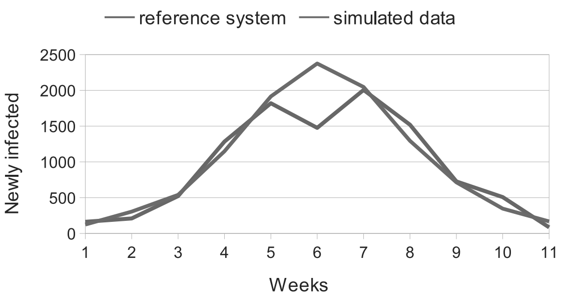


Figure 7: Results of a calibration of the influenza model using an evolutionary algorithm, 4 agent count updates and geometrical growth.

annealing with a cooling schedule adapted to the increase in the number of agents. Furthermore, the optimal number of updates needs to be determined. This number may depend on the simulation model used. In this paper, the strategy has only been applied to one type of agent-based simulation model. It needs to be tested if the results are similar with another type of model and what factors are beneficial to a good performance of this calibration method.

References

[1] Macal C, North M. Tutorial on Agent-Based Modeling and Simulation PART 2: How to Model with Agents. Proceedings of the 2006 Winter Simulation Conference. 2006; pp. 73–83.

[2] Siebers P, Macal C, JGarnet, Buxton D, Pidd M. Discrete-Event Simulation is Dead, Long Live Agent-Based Simulation! *Journal of Simulation*. 2010; 4(3):204–210.

[3] Bonabeau E. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*. 2002;99(no. Supplement 3):7280–7287.

[4] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. *SCIENCE*. 1983; 220(4598):671–680.

[5] Cerny V. Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm. *Journal of Optimisation Theory Application*. 1985; 45(1):41–51.

[6] Holland JH. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. USA: U Michigan Press. 1975.

[7] Rechenberg I. *Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Deutschland: Frommann-Holzboog Verlag. 1973.

[8] Kong C, McMahon P, Gazelle G. Calibration of Disease Simulation Model Using an Engineering Approach. *Value in Health*. 2009;12(4):521–529.

[9] Goldberg D, Deb K. A comparative analysis of selection schemes used in genetic algorithms. *Foundations of genetic algorithms*. 1991;1:69–93.

EUROSIM 2019

9th EUROSIM Congress on Modelling and Simulation

La Rioja, Logroño, Spain, July 2019

EUROSIM Congresses are the most important modelling and simulation events in Europe. For EUROSIM 2019, we are soliciting original submissions describing novel research and developments in the following (and related) areas of interest: Continuous, discrete (event) and hybrid modelling, simulation, identification and optimization approaches. Two basic contribution motivations are expected: M&S Methods and Technologies and M&S Applications. Contributions from both technical and non-technical areas are welcome.

Congress Topics The EUROSIM 2019 Congress will include invited talks, parallel, special and poster sessions, exhibition and versatile technical and social tours. The Congress topics of interest include, but are not limited to:

Intelligent Systems and Applications
Hybrid and Soft Computing
Data & Semantic Mining
Neural Networks, Fuzzy Systems & Evolutionary Computation
Image, Speech & Signal Processing
Systems Intelligence and Intelligence Systems
Autonomous Systems
Energy and Power Systems
Mining and Metal Industry
Forest Industry
Buildings and Construction
Communication Systems
Circuits, Sensors and Devices
Security Modelling and Simulation

Bioinformatics, Medicine, Pharmacy and Bioengineering
Water and Wastewater Treatment, Sludge Management and Biogas Production
Condition monitoring, Mechatronics and maintenance
Automotive applications
e-Science and e-Systems
Industry, Business, Management, Human Factors and Social Issues
Virtual Reality, Visualization, Computer Art and Games
Internet Modelling, Semantic Web and Ontologies
Computational Finance & Economics

Simulation Methodologies and Tools
Parallel and Distributed Architectures and Systems
Operations Research
Discrete Event Systems
Manufacturing and Workflows
Adaptive Dynamic Programming and Reinforcement Learning
Mobile/Ad hoc wireless networks, mobicast, sensor placement, target tracking
Control of Intelligent Systems
Robotics, Cybernetics, Control Engineering, & Manufacturing
Transport, Logistics, Harbour, Shipping and Marine Simulation

Congress Venue / Social Events The Congress will be held in the City of Logroño, Capital of La Rioja, Northern Spain. The main venue and the exhibition site is the University of La Rioja (UR), located on a modern campus in Logroño, capital of La Rioja, where 7500 students are registered. The UR is the only University in this small, quiet region in Northern Spain. La Rioja is where the Monasteries of San Millán de la Cogolla, cradle of the first words written in the Spanish language, are situated, sites included in UNESCO's World Heritage List in 1996. Of course, social events will reflect this heritage – and the famous wines in la Rioja.

CongressTeam: The Congress is organised by CAE CAE-SMSG, the Spanish simulation society, and Universidad de la Rioja.

Info: Emilio Jiménez, EUROSIM President, emilio.jimenez@unirioja.es
Juan Ignacio Latorre, juanignacio.latorre@unavarra.es

www.eurosim.info

Case Studies for a Markov Chain Approach to Analyze Agent-Based Models

Florian Kitzler*, Martin Bicher

[1] TU Wien, Institute for Analysis and Scientific Computing, Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria;
*florian.kitzler@tuwien.ac.at

SNE 27(1), 2017, 33 - 36; DOI: 10.11128/sne.27.sn.10365
Received: July 20, 2016 (Selected KA-SIM ICBTI 2015 Post-conf. Publ.), Revised: Dec. 10, 2016, Accepted: January 10, 2017
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Agent-based models have become a widely used tool in social sciences, health care management and other disciplines to describe complex systems from a bottom-up perspective. Some reasons for that are the easy understanding of agent-based models, the high flexibility and the possibility to describe heterogeneous structures. Nevertheless problems occur when it comes to analyzing agent-based models. This paper shows how to describe agent-based models in a macroscopic way as Markov chains, using the random map representation. The focus is on the implementation of this method for chosen examples of a Random Walk and Opinion Dynamic Models. It is also shown how to use Markov chain tools to analyze these models. Our case studies imply that this method can be a powerful tool when it comes to analyzing agent-based models although some further research in practice is still necessary.

Introduction

Agent-based modeling has become a widely used modeling technique and is nowadays used in many fields such as social sciences, computer science, healthcare management and economics [1]. One of the many advantages over other modeling techniques is the high flexibility of agent-based models. Another big merit is the possibility to describe heterogeneous structures. Especially these two features make it possible for agent-based models to deal with huge data sources and model very complex systems.

The availability of more powerful computational tools helps to simulate such complex models but there are still limitations when it comes to analyzing agent-based models. Very complex agent-based models have a high number of parameters and usually a lot of them are not known exactly. To parameterize or calibrate the model, a lot of simulation runs are necessary. This leads to a high amount of computational resources and time. Another big issue is the validation of the model. That means to find out if the right model was used to satisfy your needs. Hereby a problem arises as appropriate methods hardly exist for agent-based models.

The aforementioned problems underline the need for analysis-methods for agent-based models. Before we briefly explain our approach it is necessary to give short introductions to agent-based models and their connection to Markov chains. The latter poses the key-tool for our analysis method.

1 Introduction to Agent-Based Modeling and Markov Chains

In this section we will give a short introduction to agent-based models (short ABMs) and Markov chains (short MCs).

ABMs are models of systems composed of autonomous, interacting entities called agents. These agents live together in the same environment and can change their attributes after interacting with other agents or the environment. The collection of all attributes that describe the agent is called individual state of the agent.

In this work we are just regarding stochastic ABMs. This means that random effects influence the simulation results.

Stochastic models that are easier to analyze are MCs. A MC is a stochastic process that satisfies the Markov property that is also called “memorylessness” [2]. This means that the process “forgets” all about the past and the further evolution only depends on the current state. We first define the state space S of our stochastic process. That is a finite set of states our process can be in. A time-discrete stochastic process is defined as a sequence of random variables $\{X_n : n = 0, 1, 2, \dots\}$ with state space S. The variable n always stands for the time at which we observe our process. We define $p_{i,j} := \Pr(X_{n+1} = j | X_n = i)$ as the one-step transition probability from state i at time step n to state j in the next time step. The transition probabilities can be collected in a transition matrix P, where the entry in row number i and column number j correspond to $p_{i,j}$. The transition matrix P contains all information about the MC and is important to analyze its transient and asymptotic behavior.

After starting in a certain state the MC evolves following the transition probabilities. We now want to know in which state the process is when we observe it at a certain time n. Given an initial distribution vector $\mu^{(0)}$ we can calculate the distribution $\mu^{(n)}$ at time n with the formula $\mu^{(n)} = \mu^{(0)} \cdot P^n$.

The initial distribution $\mu^{(0)} = (\mu_1^{(0)}, \dots, \mu_m^{(0)})$ is given as a row vector with $\mu_i^{(0)} = \Pr(X_0 = i)$ being the probability that the MC starts in state i at time $n = 0$. So the calculation of the state distribution at time n is just n times multiplying with the transition matrix P.

Under certain conditions on the transition matrix P a unique limit distribution $\mu^* = (\mu_1^*, \dots, \mu_m^*)$ exists with $\mu_i^* = \lim_{n\to\infty} \Pr(X_n = j)$ being the asymptotic probability for the process to be in state i. Under these conditions the limit distribution can be found solving a system of linear equations, see [2].

As intended by the mentioned formulas MCs are quite simply to analyze in the contrary to ABMs. Hence a direct comparison between these techniques would improve the analysability of the ABMs.

2 Agent-Based Models as Markov Chains

This section will show that this comparison is possible as we give a step by step instruction on how to create a MC from a given ABM [3], [4].

2.1 Step1: Identifying the State Space

The first step is to find all possible states of the MC. Therefore we start with a configuration of the ABM. That is a possible combination of individual state values for each agent.

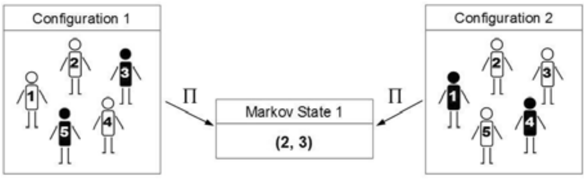


Figure 1: Two configurations that map on the same Markov state $(2, 3)$. The first number represents the number of black agents, the second number represents the number of white agents. The state space of the MC is given as $S = \{(N_b, N_w) | N_b + N_w = 5,\ N_b, N_w \geq 0\}$ with N_b ... number of black agents, N_w ... number of all white agents.

Let σ be the individual states space of an agent and N the total number of agents. We define the configuration space $\Sigma := \sigma^N$ as the set of all configurations of the ABM. Our aim is the macroscopic analysis of the model. Hence we need to find an aggregation mapping $\Pi: \Sigma \to S$ from our configuration space to a space $S \in \mathbb{N}^{|\sigma|}$. Usually this is done defining that the i-th row of $\Pi(\sigma^N)$ counts all agents with state σ_i. The image $\Pi(\Sigma)$ will furthermore pose for the state space of our MC. This idea is illustrated in Figure 1 for an ABM with only two individual states represented by the colours white and black.

2.2 Step 2: Calculate the Transition Matrix

The second step is to find all possible transitions between our Markov states and calculate the transition probabilities. Figure 2 shows the practice of how to find possible transitions.

We first start with a chosen Markov state and we want to know which state can be reached in the next time step. Then we select a configuration of the ABM that maps on our chosen Markov state. It has to be mentioned that it is generally not trivial what configuration has to be chosen if more than one of them maps on the same Markov state but in the investigated examples this is irrelevant.

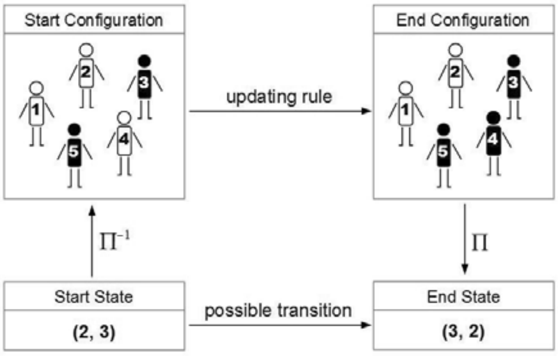


Figure 2: Schematic description of the approach of calculating the transition matrix.

With the regarded configuration we simulate one time step of the ABM using the updating rule. The updating rule contains all information about the ABM for example the movement of the agents or interacting rules. All agents update their attributes according to that rule and we observe a new configuration after the time step.

The last step is to map our new configuration again onto the state space S with Π and we have found one possible state transition. If we repeat this several times for each Markov state we can approximately calculate the transition matrix P for our MC.

Following these two steps a MC can be developed matching the macroscopic dynamics of the ABM. Hence the ABM can be investigated analysing the MC.

3 Results

We finally want to compare some results of the ABM and the corresponding MC for some test cases. Therefore we look at the transient behavior of the model at a specific time n. We fix an initial distribution and calculate the transient distribution for the MC. For the ABM we realize a Monte-Carlo-Simulation where the start values for the agents follow the same initial distribution.

3.1 1D Random Walk

We will take a look at the results of a 1D random walk model. In this case we have a 1-dimensional lattice with a finite number of sites. Furthermore a number of agents are distributed on them. Hence the only attribute of an agent is its site on the lattice. The agent can move left and right following some rules. If the agent is alone on a site he has to move on to one of the neighboring sites with the same probability. If another agent is on the same site, there is a probability of 0.6 to stay.

First we need to define the states of the MC and how the map Π works. We are considering 5 sites and 2 agents. The individual state of an agent is just an integer between 1 and 5 that holds his position on the numbered lattice. A state of the MC always contains 5 positive integers that sum up to the total number of agents, in our case to 2. If we use the symmetry of this model we can reduce the total number of Markov states to only 9.

In Figure 3 we can see the so-called transition-diagram of our MC. The arrows show all possible one-step transitions with probability greater than 0 between the Markov states represented by circles. Without knowing the exact transition probabilities we can start analyzing our process.

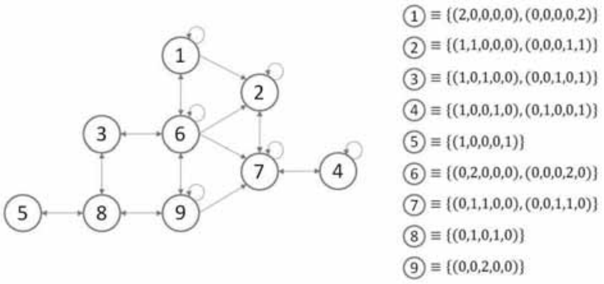


Figure 3: Transition-diagram of a 1D random walk with 2 agents that can move on 5 sites lattice. The circles represent the Markov states and the arrows represent all possible one-step transitions. On the right hand side we see the corresponding agent configurations.

We call C={2,4,7} a closed communicating class. Once the process enters one of those states, it stays in the class C forever. As C is reachable from every other state in a finite number of time steps we neglect all other states for the asymptotic behavior.

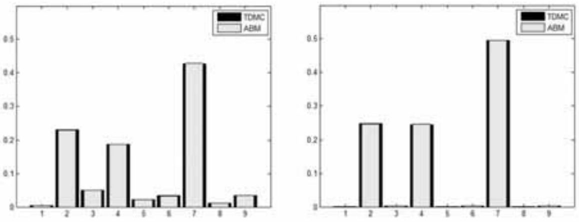


Figure 4: Comparison of the ABM (bright) with the time-discrete MC (dark). The process always starts in state 5. On the left we see the transient distribution after 10 time steps, on the right after 20 time steps. For the Monte-Carlo-Simulation of the ABM we used 100.000 iterations.

The results in Figure 4 show that the limit distribution is nearly reached after just 20 time steps. In this case we could calculate the transition probabilities by hand using the movement rules of the random walk. In our simulation the stochastic process always started in state 5 corresponding to the configuration $(1,0,0,0,1)$.

The exact limit distribution can be calculated solving a system of linear equations. It is independent of the initial distribution with $\lim_{n\to\infty} \Pr(X_n = 2) = 0.25$, $\lim_{n\to\infty} \Pr(X_n = 4) = 0.25$ and $\lim_{n\to\infty} \Pr(X_n = 7) = 0.5$.

We can also calculate the expected value of the time we first enter the closed communicating class C. When we start in state 5 this absorbing time is reached after 7.08 time steps.

3.2 Opinion Dynamics Model

Another type of model we were investigating is a two-state opinion dynamics models in which the agents are able to change their states after interacting with other agents. We were looking at different interaction network types and compared the results with regard to the Euclidean norm of the error vector. Analyzing a simulation run of the ABM with 20 agents we received a total difference to the Markov chain of $\|error\|_2 = 0.0028$.

4 Conclusion and Outlook

Our case studies show that this approach works well for the investigated examples. Using statistic tests we can show that the results of the ABM and the MC follow the same distribution. The movement and interacting models we considered can be used in more applied models as sub models. Some further research on bigger and more applied models is still needed.

Acknowledgments. K-Projekt DEXHELPP is supported by BMVIT, BMWFW and the state of Vienna via COMET – Competence Centers for Excellent Technologies. Programme COMET is processed by FFG

References

[1] Axelrod R. Agent-Based Modeling as a Bridge between Disciplines. In: Tesfatsion, L., Judd, K.L. (eds.): Handbook of Computational Economics, Agent-Based Computational Economics, Volume 2, 1565-1584

[2] Serfozo R. Probability and its Applications: Basics of Applied Stochastic Processes.

[3] Banish S, Lima R, Araújo T. Agent Based Models and Opinion Dynamics as Markov Chains

[4] Izquierdo LR, Izquierdo SS, Galán JM, Santos JI. Techniques to Understand Computer Simulations: Markov Chain Analysis. In: Journal of Artificial Societies and Social Simulation 12(1)6 http://jasss.soc.surrey.ac.uk/12/1/6.html, (2009)

Integration of Simulation-based Training for Welders

Benjamin Knoke*, Klaus-Dieter Thoben

BIBA – Bremer Institut für Produktion und Logistik GmbH, University of Bremen, Hochschulring 20, 28211 Bremen, Germany; **kno@biba.uni-bremen.de*

SNE 27(1), 2017, 37 - 44; DOI: 10.11128/sne.27.en.10366
Received: February 20, 2017 (Selected ASIM STS 2016 Postconf. Publ.), Accepted: March 15, 2017
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Simulation-based training for welders is continuously gaining importance. However, research on the integration of welding simulators into existing structures and processes is still scarce. In order to contribute towards a greater understanding, this paper collects concepts from areas with a long history of simulation-based training, such as medicine and aviation. These concepts are applied to provide a structured evaluation of two case studies that were conducted within two pioneer organisations. It was observed that certain levels of physical and functional fidelity were necessary for experienced trainers to accept welding simulators. Currently, the expected features are best provided by welding simulators based on augmented reality. Optimal results were yield in groups of 3-4 trainees, when every trainee is engaged into the simulation with a certain role (welding, correcting, filming, taking notes). Simulation-based training can be successfully applied towards a range of skills that includes technical and functional expertise training, problem-solving and decision-making skills, as well as interpersonal and communications skills.

Introduction

Vocational education and training in the field of welding is currently undergoing a transformation through the integration of simulation-based training. Although the transformation is gaining momentum from advances in multimedia technologies, the implementation of simulation-based training is still in its infancy. Much less research is available than in areas with a long history of simulation-based training, such as the medical sector.

In order to contribute towards a greater understanding of the possibilities that these simulation technologies have to offer, this paper discusses the applicability of find-ings from other sectors towards simulation-based train-ing for welders.

For an educated design and the integration of simulation technologies for welders, it is necessary to understand the benefits and relevant characteristics of training simulations. Although it seems that a training simulation simply improves the closer it comes to reality, the relation is instead more complex [1]. Not only can unnecessary details increase the costs of simulators [2] [3], they also bear the risk to divert the focus from the intended training [4]. Therefore, Shirts (1992) recommends to ‘look past the details to the essence of reality’ [4].

This paper discusses concepts that contribute to the success of simulation-based training in various sectors, such as simulation fidelity, definition of training objectives, and benefits of simulation technologies. The currently available welding simulators and two case studies are analysed to investigate the current state-of-the-art and the applicability of the aforementioned concepts into welding training.

1 Related Work

1.1 Simulation Fidelity

Early research initiated by Thorndike & Woodworth in 1901 argued that a simulated environment must have the same elements and surface features than its real counterpart in order to allow a transfer of problems [5] and to evoke engagement of the trainee [6]. This demand influenced the definition of the term ‘simulation fidelity’ in the middle of the 20th century. Simulation fidelity is used to describe the degree to which the real operational equipment or situation is resembled by a simulator [1] [2].

Subsequent studies indicate that the correlation between simulation fidelity and training effectiveness relies on multiple dimensions. Initially, Kinkade and Wheaton (1972) defined three dimensions of simulation fidelity [7]:

- *Equipment fidelity*: the degree to which the simulator looks and feels like the original operational equipment.
- *Environmental fidelity*: the degree to which the simulator resembles the sensory stimulation and control feel of the original task situation.
- *Psychological fidelity*: the degree to which the trainee perceives the simulator as a duplicate of the original operational equipment and task situation.

This definition has been refined by Fink and Shriver (1978), who defined two key dimensions [8]:

- *Physical fidelity* as the degree to which a simulation represents the appearance and feel of the original equipment (previously: equipment fidelity*)*.
- *Functional fidelity* as the degree to which the original stimulus and response options are implemented in the equipment (previously: environmental fidelity*)*.

Hays (1980) emphasises the use of the terms physical and functional fidelity [9]. He also states that psychological fidelity and corresponding approaches can be derived from physical and functional fidelity and should therefore be discarded. Although physical fidelity and functional fidelity also show interdependencies, the functional fidelity of a training simulation is mainly considered to determine its performance [10] [11].

Although the term fidelity is widely used since its introduction, Hamstra et al. (2014) argue that the differentiation between functional and physical dimensions is not made consistently across literature [12]. The authors also emphasise the importance of functional task alignment (functional fidelity) over physical resemblance (physical fidelity), which is always determined in context of the underlying instructional goals [12].

The knowledge to differentiate between important and unnecessary fidelity is one of the key objectives during the design of simulation-based training, because minimal costs can only be achieved by selecting just the fidelity that is necessary to meet the training objectives [13].

The conclusions drawn for the methodology of this paper are (i) to distinguish between physical from functional fidelity, (ii) to analyse if the importance of functional over physical fidelity can be confirmed for welding simulators, and (iii) to benchmark functional fidelity in consideration of specific training objectives. Approaches to describe the skills that can be achieved through simulation-based training are described in the following section.

1.2 Skills that can be trained through simulation-based training

The main objective of training is the acquisition, improvement or testing of skills [14] [15] [16]. The skills that can be trained through simulations have been classified by Lateef (2010) into three main types [16]:

- Technical and functional expertise training.
- Problem-solving and decision-making skills.
- Interpersonal and communications skills or team-based competencies.

A similar categorization has been published by Larnpotang et al. (2013), who defined a skills triangle to emphasise the possible training of multiple dimensions in a single training simulation, as shown in Figure 1 [17].

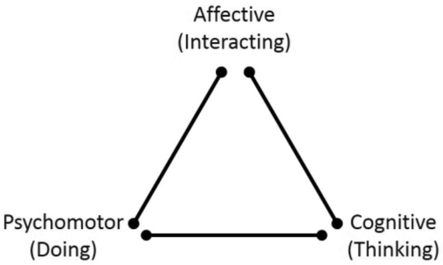


Figure 1: The skills triangle of simulation-based training [17].

In addition to an improved acquisition of skills, higher fidelity in simulation-based training can offer several benefits in comparison to regular training.

1.3 Features and uses of fidelity in simulation-based training

Issenberg et al. (2005) conducted an extensive literature review on the features and uses of successful high-fidelity medical simulations with the following results, sorted by weight [18]:

- Feedback is provided during learning experience
- Learners engage in repetitive practice
- Simulator is integrated into overall curriculum
- Learners practice with increasing levels of difficulty
- Adaptable to Multiple Learning Strategies
- Capture Clinical Variation
- Operate in a controlled Environment
- Individualised Learning
- Defined Outcomes and Benchmarks
- Simulator validity

Although the study has been conducted in the medical field, the results are expected to be applicable as a framework to examine the features and uses of fidelity in training simulations of welders.

1.4 Simulation technologies in welding simulators

Simulation technologies can be defined as materials and devices created or adapted to solve practical problems and create simulations [19]. The current key simulation technologies in welding are virtual and augmented reality. During the past decade these technologies were applied to create welding simulators and made the transition towards practical application [20] [21]. The commercial solutions that are currently available are listed in Table 1.

Manufacturer	Product	Characteristics
GSI SLV Halle	GSI SLV Halle Schweißtrainer	Real low-power arc for melt run, requires shielding gas
Seabery	Soldamatic	Augmented Reality over an artificial workpiece
Fronius International	Virtual Welding	Virtual Reality, artificial workpiece in fixed position for haptic feedback
Lincoln Electric	VRTEX	Virtual Reality, artificial workpiece in fixed position for haptic feedback
123 Certification	ARC+	Virtual Reality, no haptic feedback
EWM Hightec Welding	EWM Virtual Welding Trainer	Virtual Reality on screen, no headgear, no haptic feedback

Table 1: Overview on available welding simulators.

All of the listed simulators rely on optic measurement to capture the position and movement of a welding torch or electrode. The measured characteristics include stick out, work angle, travel angle, travel path, and travel speed.

The listed welding simulators differ greatly in physical and functional fidelity, as well as in price. Currently, the augmented reality system (Soldamatic) seems to be dominant, which has also been observed within the two case studies that are described in the following section.

2 Application of Welding Simulators

Within the course of the German research project 'MESA – Medieneinsatz in der Schweißausbildung' (media applications in welding training), two diverse case studies have been conducted to analyse the implementation of welding simulators in industrial practice. The case studies include a plant for the construction of car chassis of a car manufacturer and an organisation for joint training of metal processing SMEs.

2.1 Case study 1: Car chassis manufacturing plant

The first case study was conducted at a car chassis manufacturing plant of a German car manufacturer. On average, 50 trainees periodically perform a welding training with a duration of four weeks. This includes mostly apprentices, but also advanced training sessions and individual coaching for employees.

The introduction of welding simulators started with two VRTEX-systems that were mostly used as demonstrator and had low impact on the practical training. In 2015 four Soldamatic systems have been purchased. The decision to change the system was made due to the considerably lower price and a better physical and functional fidelity of the simulation. The augmented reality system was preferred due to a more robust mode of operation and constraints of motion sickness through limited environmental perception within the virtual reality simulation. It was observed that VRTEX-systems create strong electromagnetic fields and are prone to failure, if positioned within 4-5 metres of other sources of electromagnetic fields or each other.

In collaboration with the simulator manufacturer, an individual workpiece was integrated in the simulation to enable the welding training along a curved outline on a control arm that is actually manufactured within the chassis plant and used in welding training. The control arm was replicated as a plastic workpiece with reference markers. It is used within the welding training to increase the simulators acceptance among the trainers through an increased connection to reality and to save resources.

Each simulator is used by a group of 3-4 trainees. After the initial enthusiasm fades, it is considered important to engage every trainee in the simulation through a specific task. While one trainee welds within the simulation, the second trainee analyses an external screen of the simulation and communicates corrections. The third trainee films the process and focusses on ergonomic aspects. To make the posture more visible, a white adhesive strip is vertically taped on the back of the first trainee. If present, an optional fourth trainee is instructed to fill an evaluation sheet.

After an initial scepticism and fear for their occupational safety, the simulators are now greatly appreciated by all welding trainers. They see the simulators as an easy way to provide individual feedback that is not contested by the trainees. To further increase the acceptance among experienced welding trainers, it is important to state that the simulation should not be seen as a perfect replication. The physical fidelity is lowered through characteristics such as a low travel speed that is considered to be more like ‘Tai-Chi’ by the trainers. However, the trainers generally observe a ‘significant learning effect’ through the simulation-based training, implicating a high functional fidelity.

2.2 Case study 2: Joint training organisation

The second case study concerns a regional joint training organization that trains apprentices and experienced employees for 65 metal processing SMEs in central Germany.

The practical welding training is usually conducted in groups of three trainees. In addition to the 15 welding booths, the organization purchased one ARC+-system in 2013 and also one Soldamatic-system in 2016. Complaints were made that the ARC+-system is rather fragile and had to be sent to Canada multiple times for repairs. Also it allows less customization than the Soldamatic-system and provides no haptic feedback through lack of a physical workpiece.

The simulators have been integrated into basic welding training for beginners. The objectives were to ensure an ergonomic posture, train the correct parameters, such as travel speed and work angle, and to generate direct and individual feedback. The simulation-based training resulted in ‘much better results’ during the trainees’ first workpieces and a focus on ergonomic aspects that was not possible in the conventional welding booths, due to blinds and personal safety gear. The welding tasks can be reset quickly and do not require the preparation of workpieces, leading to much faster exercises and a steeper learning curve. A time saving of up to 40-50% through a combination of simulation-based and conventional training was observed to teach basic welding skills to beginners. The trainers also noticed an increased level of engagement during the group sessions in simulation-based training. This appears to be linked to the gamification aspect of the simulations, as trainees contest each other for higher scores.

In addition to training, the mobile design and operational safety of the simulators also allows to use them to attract attention on local trade fairs and in events for occupational orientation.

3 Discussion of Results

During the following discussion, the concepts described in Section 2 are evaluated in consideration of results from the two reported case studies.

3.1 Fidelity requirements

A certain level of *physical fidelity* appears to be beneficial in order to support the integration of simulators into existing structures and processes. The case studies showed that an experienced welding trainer is more likely to approach a simulator that resembles traditional welding equipment. Therefore, the processing equipment of the most successful welding simulators is integrated into cases of traditional welding equipment and applies a similar set of controls.

Although the welding simulators perform well for their intended purpose, an experienced welder requires 3-5 ‘test runs’ in order to generate good results on a welding simulator. This shows a certain lack of *functional fidelity* that is usually accepted, if the difference between simulation and replication is explained to the welder. The level of acceptance is higher if the simulator is seen as equipment that allows to train skills that be transferred into welding.

3.2 Skills that can be trained through welding simulators

The positive learning effect through (partial) integration of simulators in the training of welders has been proven in multiple studies [22] [23] [24] [25] and was also observed in both case studies. The skills that can be trained through welding simulators are described in the following along the three main categories [16].

Technical and functional expertise. Technical and functional expertise comprises the main objective of practical welding training. The key parameters that are monitored by all welding simulators are stick out, work angle, travel angle, travel path, and travel speed.

In addition to the parameters that impact the weld, welding simulators have shown great potential in training of an ergonomic posture. Instead of isolated training in welding booths, where welding apprentices are mostly rated by their results, simulation-based training can be conducted under supervision or in groups to put the focus on the process. Currently, ergonomic training is not directly implemented in welding simulators, but both case studies showed great interest and applied work-arounds. Group training and video recording can be applied to create awareness for posture during welding training.

Problem-solving and decision-making skills. The current state-of-the-art welding simulators create complex simulations that depend on various parameters, such as voltage, current, shielding gas type and flow rate. These parameters have to be configured for training sessions and trainees can also experiment with the outcomes of parameter manipulation without suffering serious safety constraints.

The simulations can be applied to provide a link from theory to practice and support classroom situations through combination with an external video projector, e. g. in vocational training schools.

Most simulators include some ability to display learning material and conduct tests. However, this function has not been used within the case studies, because the simulators are usually operated in groups of 3-5 trainees, which contradicts traditional test situations.

Interpersonal and communications skills. In comparison to traditional training, the simulators are usually used in small groups of trainees. Within the chassis manufacturer plant, each group member was given a specific task during the individual training sessions (welding, correcting, filming, taking notes), which led to an increase in communication and team-work. Both case-studies also reported informal competitions between the trainees as a positive effect that increases engagement.

Furthermore, the welding sector is characterised by a relatively high level of cultural diversity, which can cause language barriers. Visualisations and multi-language support are implemented in most simulators and can be applied to facilitate the teaching of technical terms.

Overall, skills in all three categories are impacted by training with welding simulators. However, interaction and communication are merely seen as a corollary of psychomotor and cognitive skills.

3.3 Features and uses of fidelity in simulation-based training

The features and uses of fidelity in simulation-based training that have been defined by Issenberg et al. (2005) within the medical sector [18] are discussed in consideration of the characteristics of welding simulators and the performed case studies.

Feedback is provided during learning experience. All of the currently successful welding simulators include feedback concerning the monitored parameters that can be shown during welding sessions. They also feature an evaluation screen that shows the course of parameters along the weld after completion.

In combination with group work, the feedback feature was considered highly important in both use cases and is expected to be a main cause for skill improvement.

Learners engage in repetitive practice. The ability to quickly restart a welding session is seen as a great advantage over traditional training, which would require preliminary work, such as cutting and grinding of the workpiece. Instead, simulation-based training sessions can be repeated under identical circumstances within the press of a button.

Simulator is integrated into overall curriculum. Multiple cases of successful implementation of welding simulators in vocational training are existing. However, the curriculum integration is still in a preliminary state and differs greatly between organisations in scope and evaluation.

The MESA-project is currently performing research towards a curriculum integration of welding simulators and cooperates with the German Welding Society (DVS) to develop a structured guideline.

Learners practice with increasing levels of difficulty. Implemented difficulty levels in welding simulations vary through more or less strict evaluation, variation of tasks, or (partially) turned off feedback during the exercise. The variation of feedback was considered helpful, but not necessary during the case studies. It has been characterised as a viable approach to further prepare beginners in advance of their first actual welding training.

Adaptable to Multiple Learning Strategies. Most welding simulators can be used in learning situations of single users or groups with or without the presence of instructors. The simulation is usually displayed within a head-mounted display that resembles a welding helmet, and a small screen that is placed on the processing unit. Through connection of external video equipment, the simulations can also be used in classroom settings.

Capture (Clinical) Variation. The high variation of products and processes that confront welders in industrial practice are difficult to transfer into simulation-based training and seem to be a general barrier for simulation-based training in production.

An integration of individual products showed to have a positive effect on the acceptance of simulation-based training, and to reduce waste during the production of those products. During the case study of the car chassis manufacturer, considerable effort was spent to integrate a specific control arm into the augmented reality simulation of the Soldamatic-system. However, a low-cost or short-term variation of workpieces remains impossible, as long as the systems rely on digital representations of workpieces and an on-the-fly digitalization is not implemented.

A promising feature of augmented reality-based simulators is that the simulation is integrated into the real environment. During the simulation of MAG welding, the Soldamatic-system enforces a certain position and orientation of the headgear and welding torch towards the workpiece. This allows to mount the workpiece into confided spaces or underneath a table to simulate extraordinary processes (e. g. Figure 2), such as the welding of pressure tanks or heat exchangers.

This can be applied to provide ergonomic assistance during constrained postures and to improve confidence towards complex task or unanticipated events (as in [16]).

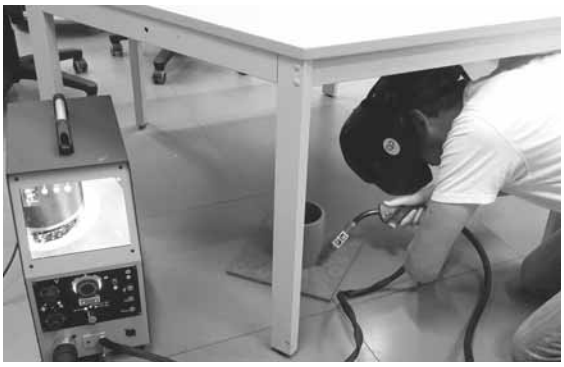


Figure 2: Welding simulation in confined spaces through Augmented Reality.

Operate in a controlled Environment. Welding simulators provide the possibility to train in groups and to freely variate process characteristics, which is usually not possible in conventional training due to safety constraints. Within the case studies, welding trainers expressed great delight towards the opportunity to provide support during the process and to improve consciousness for an ergonomic posture.

Individualised Learning. Individual requirements can be met through variation of time spent in the simulation and through adjustment of difficulty levels. Additionally, the simulators provide several sets of standard tasks with the most prominent types of manual welding processes: MIG, GTAW, and SMAW.

Defined Outcomes and Benchmarks. The task and benchmarks are clearly defined by all observed welding simulators and the monitored parameters are graded in percent that were achieved during the training sessions. However, beginners require a short introduction to the simulation to correctly interpret the provided feedback. A suitable tutorial is missing.

Also, welding trainers reported the neutral feedback through sensor data as a strong improvement over traditional training, where their feedback is sometimes either not taken seriously or causes unnecessary strain.

Simulator validity. The validity of welding simulators corresponds with their functional fidelity and has been proven in both use cases. System resilience was described as an important criteria by both use cases for their decision to purchase specific welding simulators.

4 Conclusion and Outlook

During the past decade, simulation-based training for welders continuously gained importance. Since extensive research on the integration of simulators has already been conducted in other areas, such as medicine or aviation, this paper reviewed established concepts to evaluate two case studies on the integration of welding simulators that were performed in pioneer organisations.

The evaluation proved that welding simulators are greatly appreciated by welding trainers, once certain requirements are met. The expectations towards physical fidelity mostly address intuitive controls that simulate those of conventional welding equipment. It was also observed to be important that welding simulators are presented as training equipment and not as a perfect replication of reality. Otherwise, experienced welders, who usually do not perform well on their first tries, tend to get upset and focus on system flaws.

In accordance with previous research, both case studies reported a 30-50% decrease in time that is necessary to develop basic welding skills, when simulators are integrated into welding training for beginners. Best results were achieved during simulation-based training in groups of 3-4 persons, when each person was engaged into the simulation through an individual task (welding, correcting, filming, taking notes), or during individual training with an instructor.

Most features and uses of fidelity that contribute towards the success of simulations in the medical sector, also apply for welding simulators. Though, the individuality of products and processes seems to be a general barrier towards the customization of simulations in production. A demand of customization showed, as one of the two organisations spent considerable effort to integrate a specific workpiece into a welding simulator.

A promising application of augmented reality is the opportunity to fit the available standard workpieces into confined spaces. This allows the simulation of complex welding tasks and the support of ergonomic postures with relatively low effort. Following research may include experiments to transfer welding simulations into specific environments, such as pressure tanks or heat exchangers.

Acknowledgment

This work has been funded by the German Federal Ministry of Education and Research (BMBF) through the research project MESA – Medieneinsatz in der Schweißausbildung (media applications in welding training, http://mesa-projekt.de). The authors wish to acknowledge the ministry for their support. We also wish to acknowledge our gratitude and appreciation to MESA project partners for their contribution during the development of various ideas and concepts presented in this paper.

References

[1] Miller GG. *Some considerations in the design and utilization of simulators for technical training (AFHRL TR-74-65)*. Brooks Air Force Base: Air Force Human Resources Laboratory; 1974.

[2] Hays RT, Singer MJ. *Simulation fidelity in training system design: Bridging the gap between reality and training*. New York: Springer Science & Business Media; 1989.

[3] Holding DH. *Transfer of training.* In: Morrison, JE. *Training for performance: principles of applied human learning.* New York: John Wiley; 1991. pp 93-125.

[4] Shirts RG. 10 Secrets of Successful Simulations. T*raining*. 1992; 29(10): pp. 79-83. 1992.

[5] Thorndike EL, Woodworth RS. The influence of improvement in one mental function upon the efficiency of other functions. *Psychological Review*. 1901; 8(3): pp. 247-261. doi: 10.1037/h0074898.

[6] Bradley P. The history of simulation in medical education and possible future directions. *Med Education*. 2006; 40(3): pp. 254–262. doi: 10.1111/j.1365-2929.2006.02394.x.

[7] Kinkade RG, Wheaton GR. *Training device design*. In: Van Cott HP, Kinkade RG. *Human engineering guide to equipment design*. Washington: US Government Printing Office; 1972. pp. 668-699.

[8] Fink CD, Shriver EL. *Simulators for maintenance training: Some issues, problems and areas for future research*. Alexandria: Kinton Inc; 1978.

[9] Hays RT. *Simulator fidelity: A concept paper (No. ARI-TR-490)*. Alexandria: Army Research Institute for the Behavioral and Social Sciences; 1980.

[10] Allen JA, Hays RT, Buffardi LC. Maintenance training simulator fidelity and individual differences in transfer of training. *Human Factors*. 1986; 28(5): pp. 497-509. doi: 10.1177/001872088602800501.

[11] Davidovitch L, Parush A, Shtub A. The impact of functional fidelity in simulator-based learning of project management. *International Journal of Engineering Education*. 2009; 25(2): pp. 333-340.

[12] Hamstra SJ, Brydges R, Hatala R, Zendejas B, Cook DA. Reconsidering fidelity in simulation-based training. *Academic Medicine*. 2014; 89(3): pp. 387-392. doi: 10.1097/ACM.0000000000000130.

[13] Fletcher JD. Using networked simulation to assess problem solving by tactical teams**.** *Computers in human behaviour*. 1999; 15(3): pp. 375-402. doi: 10.1016/S0747-5632(99)00028-X.

[14] Gaba DM, Howard SK, Flanagan B, Smith BE, Fish KJ, Botney R. Assessment of clinical performance during simulated crises using both technical and behavioral ratings. *The Journal of the American Society of Anesthesiologists*. 1998; 89(1): pp. 8-18, 1998. doi: 10.1097/00000542-199807000-00005.

[15] Gupta A, Peckler B, Schoken D. Introduction of hi-fidelity simulation techniques as an ideal teaching tool for upcoming emergency medicine and trauma residency programs in India. *Journal of emergencies, trauma, and shock*. 2008; 1(1): pp. 15-18. doi: 10.4103/0974-2700.41787.

[16] Lateef F. Simulation-based learning: Just like the real thing. *Journal of Emergencies, Trauma, and Shock*. 2010; 3(4): pp. 348-352. doi: 10.4103/0974-2700.70743.

[17] Larnpotang S, Lizdas D, Rajon D, Luria I, Gravenstein N, Bisht Y, Schwab W, Friedman W, Bova F, Robinson A. Mixed simulators: augmented physical simulators with virtual underlays. *Virtual Reality (VR)*; 2013 March; Orlando: IEEE. pp. 7-10. doi: 10.1109/VR.2013.6549348.

[18] Issenberg SB, Mcgaghie WC, Petrusa ER, Gordon DL, Scalese RJ. Features and uses of high-fidelity medical simulations that lead to effective learning: a BEME systematic review. *Medical teacher*. 2005; 27(1): pp. 10-28. doi: 10.1080/01421590500046924.

[19] Cook DA, Hatala R, Brydges R, Zendejas B, Szostek JH, Wang AT, Erwin PJ, Stanley J, Hamstra SJ. Technology-enhanced simulation for health professions education: a systematic review and meta-analysis. *JAMA*. 2011; 306(9): pp. 978-988. doi: 10.1001/jama.2011.1234.

[20] Fast K, Gifford T, Yancey R. Virtual training for welding**.** *Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2004)*; 2004 Dec; Arlington: IEEE. pp. 298-299. doi: 10.1109/ISMAR.2004.65.

[21] White S, Prachyabrued M, Baghi D, Aglawe A, Reiners D, Borst C, Chambers T. Virtual welder trainer. *2009 IEEE Virtual Reality Conference*; 2009 March; Lafayette: IEEE. pp. 303-303. doi: 10.1109/VR.2009.4811066.

[22] Stone RT, Watts KP, Zhong P, Wei CS. Physical and cognitive effects of virtual reality integrated training. *Human Factors*. 2011; 53(5): pp. 558-572. doi: 10.1177/0018720811413389.

[23] Wang Y, Nan Z, Chen Y, Hu Y. Study on welder training by means of haptic guidance and virtual reality for arc welding. *International Conference on Robotics and Biomimetics (ROBIO'06)*; 2006 Dec; Kunming: IEEE. pp. 954-958. doi: 10.1109/ROBIO.2006.340349.

[24] Okimoto MLL, Okimoto PC, Goldbach CE. User Experience in Augmented Reality Applied to the Welding Education. *Procedia Manufacturing*. 2015; 3: pp. 6223-6227. doi: https://doi.org/10.1016/j.promfg.2015.07.739.

[25] Mavrikios D, Karabatsou V, Fragos D, Chryssolouris G. A prototype virtual reality-based demonstrator for immersive and interactive simulation of welding processes. *International Journal of Computer Integrated Manufacturing*. 2006; 19(3): pp. 294-300. doi: 10.1080/09511920500340916.

Physical Simulation Related Exercises for the Education in the STEM Field – Approaches Based on the Physolator Framework

Dirk Eisenbiegler[1*], Dietmar Gruber[2], Thomas Jörg[2]

[1]University of Furtwangen, Germany; *dirk.eisenbiegler@hs-furtwangen.de

[2]Hector Seminar, Germany

SNE 27(1), 2017, 45 - 52; DOI: 10.11128/sne.27.en.10367
Received: February 20, 2017 (Selected ASIM STS 2016 Postconf. Publ.), Accepted: March 10, 2017
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. This paper presents different kinds of approaches towards using physical simulation based exercises for early teaching of STEM topics at school[1]. The approaches presented in this paper are based on the Physolator physics simulation framework. This paper analyzes to which degree these approaches can be used to meet given teaching goals.

Introduction

In our world, computer based physical simulations are omnipresent. They are used for science and engineering as well as in computer games and in animations for the movie industry. Professionals working with computer based physical simulations are aware of the fact that it takes good skills at least in the following fields for producing such computer based applications: physical modeling, geometry, numerical mathematics, programming and graphics programming. The significant skills for producing physical simulations are all located in the STEM fields[1]. Work in this field is interdisciplinary. It takes physicists, mathematicians and computer scientist working together in such projects.

The guiding questions of this paper: Could physical simulation be used in early teaching of mathematics, physics and computer science at school? Could this lead to more application oriented understanding of the different topics in mathematics, physics and computer science? Could this strengthen an interdisciplinary thinking for these domains?

1 Teaching Physics and Physical Simulation

There are numerous publications dealing with the question on how to improve teaching in science and especially in physics [7,8,9]. In a physics lecture, students shall learn the theory of physics and they shall learn how apply this knowledge to real world scenarios. Experiments play an important role when teaching physics. Experiments are used to confirm theoretical models. On the other hand side, experiments are stimulated by theory. Students should learn, how theoretical models are used to describe nature and how experiments in nature are used to verify theoretical models.

Physical simulation is a supplement to theory and experiment. Working with physical models means applying theoretical knowledge [6]. Just like real world experiments, physical simulations are motivated by theory. The students have to analyze the simulation runs to see if or if not the physical simulation confirms their expectations about the real world behavior. Physical simulations can never be a substitute for experiments. Physical simulations are used for verification and clarification: Is the physical simulation, which is based on the theoretical formulas, consistent with the real world observations from experiments?

A learner who is starting to set up his first physical simulation is confronted mainly with two different challenges: Understanding the physical model and understanding the way of implementing the physical model using a programming language.

[1] STEM = science, technology, engineering, and mathematics

The following aspects have to be considered when learning programming: learn the syntax of a programming language, learn the fundamental, imperative concepts of programming and optionally learn the concepts of object orientated programming such as classes and instances, inheritance and dynamic binding. From a didactic point of view it is important to keep the different aspects of programming as separated as possible and to teach them in a step-by-step manner.

The approaches presented in this paper are based on the Physolator framework (see www.physolator.de). Physolator uses the Java programming language. Java is taught in many secondary schools. Frameworks such as Mathematica or Matlab could also be used for physical simulations. However, they come with their own programming language and the student has to learn this extra programming language to get started with such frameworks. The Physolator is Java based. Students do not have to learn an extra programming language before they get started with physical simulations.

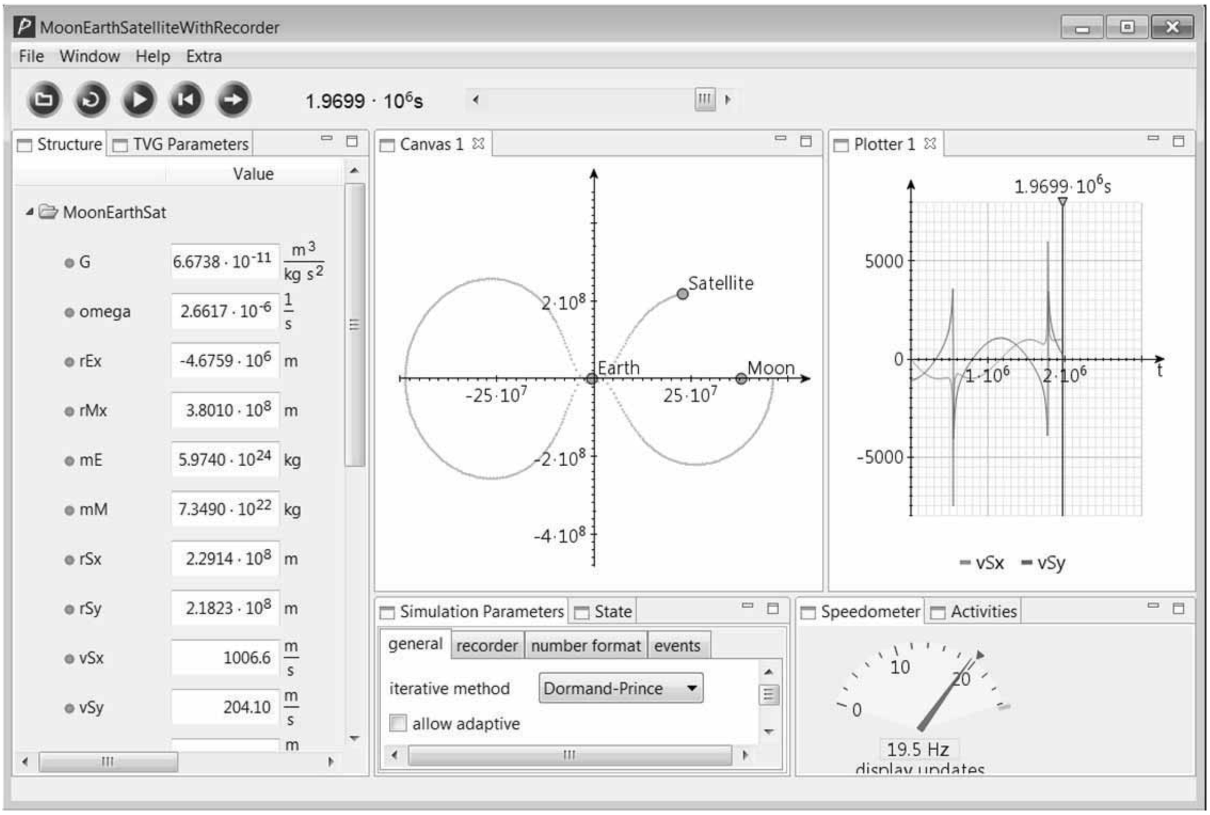
Applying Java in the context of physical simulations helps students to acquire a deeper understanding of Java since implementing a physical simulation is an exercise of already taught programming lessons.

Furthermore, Physolator is able to encapsulate the higher level OOP parts of Java, therefore entanglements of too many didactic aspects can be avoided effectively.

An exercise on the beginner level should put a focus on one topic only: One exercise for learning the meaning of gravity by playing with a given model, another exercise for learning how to build a physical model, another exercise for learning how to build graphics components. Each of these topics should first be learned in an isolated way. When for example building a first physical model in a science class, students should put a focus on understanding the underlying physical context, and the corresponding formulas. Students should not at the same time have to deal with the complexity of modern object oriented programming or with numerical mathematics or with graphics programming.

2 The Physolator Framework

Programming a physical simulation from scratch is considered to be very challenging. The Physolator is designed for physical simulations at the beginner's level. Physical systems are implemented as Java programs.



$1.9699 \cdot 10^6$s

	Value	
MoonEarthSat		
G	$6.6738 \cdot 10^{-11}$	$\frac{m^3}{kg\ s^2}$
omega	$2.6617 \cdot 10^{-6}$	$\frac{1}{s}$
rEx	$-4.6759 \cdot 10^{6}$	m
rMx	$3.8010 \cdot 10^{8}$	m
mE	$5.9740 \cdot 10^{24}$	kg
mM	$7.3490 \cdot 10^{22}$	kg
rSx	$2.2914 \cdot 10^{8}$	m
rSy	$2.1823 \cdot 10^{8}$	m
vSx	1006.6	$\frac{m}{s}$
vSy	204.10	$\frac{m}{s}$
		m

iterative method: Dormand-Prince

□ allow adaptive

19.5 Hz
display updates

Figure 1: The Physolator Framework.

To build your own physical system up and running, you write a piece of Java code, load it to the Physolator framework and then start the simulation by pressing the start button inside the Physolator. The Physolator framework is based on ODE solvers. From the Physolator's perspective, physical values are initial value problems and it uses ODE solvers for executing the simulations. Besides the ODE solvers, the Physolator framework also supports an event oriented programming style for simulating physical events such as collisions.

Figure 1 shows a snapshot of the Physolator framework. A physical system with a satellite revolving around moon and earth has already been loaded. Physical variables are always displayed on the left. Their values change during simulation. On the middle there is a graphical representation of the physical system and on the right there is a plotter for displaying the function graphs of selected variables. With the round buttons on the top you can load physical systems, start and stop them. During simulation time the user may also interfere and change the variables values manually. Editing the variables immediately changes the state of the system and its graphical representation.

The Physolator has been designed for modular style of implementing physical systems. With the Physolator one can build a set of basic physical components. Using this set of physical components, one can build more complex components by just joining together the basic components. Also the graphical components for visualizing the simulation results as well as the numerical procedures are developed independently. For every exercise the instructors provide the students with a set of base components: physical components, graphics components and numerical procedures. During the exercise, the students have to build a physical system on top of this infrastructure. They can focus their work on a very specific task. The given infrastructure of components should cover all the aspects the students should not have to take care off.

This paper presents different categories of exercises related to physical simulation with a focus on exercises at the beginner level. The different categories represent different categories of didactic concepts.

With the Physolator one could also define exercises for advanced students such as programming numerical algorithms. Such advanced tasks, however, are not part of the scope of this paper. This paper limits itself to exercises that are well suited for students at school level or at the beginner level at a university.

2.1 Category 1: Experimentation with a given physical system

Traditionally, physical experiments are based on mechanical or electrical devices. During the exercise, the students build an apparatus by assembling these devices and then execute different runs with varying parameters and conditions. Such experiments pursue the following teaching goals:

- Give the students a practical experience of the theoretical physical concepts presented during the theoretical parts of the lecture (e.g. gravitation, friction...).
- Make sure, the students can apply theoretical concepts to the real world: relationship between variables and formulas in the theoretical world and observations in the real world.
- Explain to the students the meaning of physical models computer based simulations as a part of the scientific research process of physicists.

Readymade computer simulations can be a complement for such real experiments. For good real world examples you need the right devices. Some of them are costly and assembling an apparatus is time consuming. Some experiments that would be useful to for a better understanding of the physical domain, simply cannot be run in a classroom. One can simulate the orbit of a satellite on the computer, but not in a classroom.

The moon-earth-satellite example is a physical system that is well suited exercises of this category. Exercises: Give the satellite the right initial position and speed and observe the path of its movement! Try find an initial position and speed so the satellite runs on a closed orbit! During simulation time observe the relevant forces and accelerations: gravity, Coriolis force, centrifugal force!

In this kind of experiment, the student loads a readymade physical system to the Physolator. In this virtual experiment interacting with the physical system means ‘playing’ with the physical variables and observing the impact on the behavior of the physical system.

2.2 Category 2: Building a physical model

In this kind of experiments, mathematical formulas shall be used to build a model of a simple physical system. The students shall write down the physical variables and physical formulas in Java notation and then load and start their physical system.

The learning tasks to be pursued with this kind of exercises are:

- Learn, how to describe a consistent physical model using physical variables, formulas, and derivation relationships and provide the physical system with an initial state.
- Understand the relationship between formulas defining the behavior and temporal progressions of physical systems following the rules defined by these formulas.
- Learn how to define physical models and learn about the limitations of physical models.

In this kind of exercise, students write Java program code. The programming language Java, however, is used in a very limited way. In this context it is only used for writing down physical variables and formulas. The entire program code only consists of Java variable declarations and value assignments to variables. The Java variables correspond to physical variables. Java variables are a means for representing physical variables in a computer. Variable assignments are a means for representing formulas. The variable assignment in Java assigns a value to the variable. The value is defined by a mathematical expression. The mathematical expression represents the formula.

The following program code gives an example for such a physical system.

```
public class SpringMassPendulum
           extends PhysicalSystem {
  @V(unit = "m")
  public double p0 = 1;
  @V(unit = "m")
  public double g = -9.81;
  @V(unit = "kg")
  public double m = 0.03;
  @V(unit = "N/m")
  public double D = 2;
  @V(unit = "")
  public double k = 0.05;
  @V(unit = "N")
  public double F;
  @V(unit = "m", derivative = "v")
  public double x = 0;
  @V(unit = "m/s", derivative = "a")
  public double v;
  @V(unit = "m/s^2")
  public double a;
  public void f(double t, double h) {
    F = g * m + D * (p0 - x) - k * v;
    a = F / m;
  }
  public void initPlotterDescriptors(
          PlotterParameters r) {
    r.add("x,v,F", 5, -7, 7);
  }
}
```

The program code above defines a spring mass pendulum with damping. The mass *m*, the pivot position *p0*, the earth acceleration *g*, the spring constant *D* and the coefficient of friction *k* are given constants. The position of the point mass *x* and its actual velocity *v* define the state of the physical system. Annotations *@V* are used for attaching physical units to the variables and for defining derivation relationships. In this case, *v* is the first derivative of *x* and *a* is the first derivative of *v*. The formulas are defined inside method *f*. These formulas define the actual force and the actual acceleration.

This kind of Java program represents a physical system. The students learn, that this kind of notation is used for writing down physical variables, formulas and derivation relationships. At that time, the students do not necessarily have to understand Java. There are no concepts being used that go beyond physical variables and formulas – no control structures, no parameters, no methods, no exception handling etc..

The initPlotterDescriptors method declares, that *x*, *v* and *F* shall be plotted during simulation. Figure 2 shows the result of the simulation run.

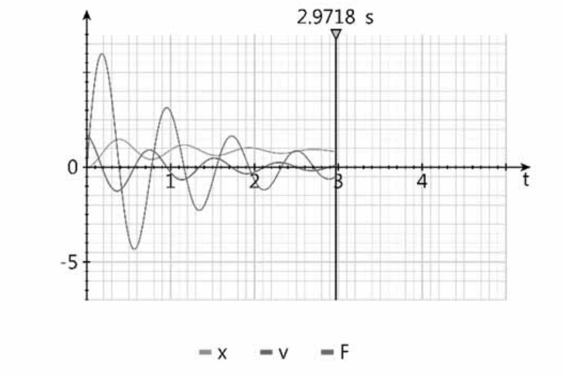


Figure 2: Spring mass pendulum with damping.

2.3 Category 3: Building a physical system by composing given physical components

In this kind of experiment, physical systems are build up by composing given physical components. The following program code describes a double pendulum with two point masses connected to a pivot point via two springs (see Figure 3).

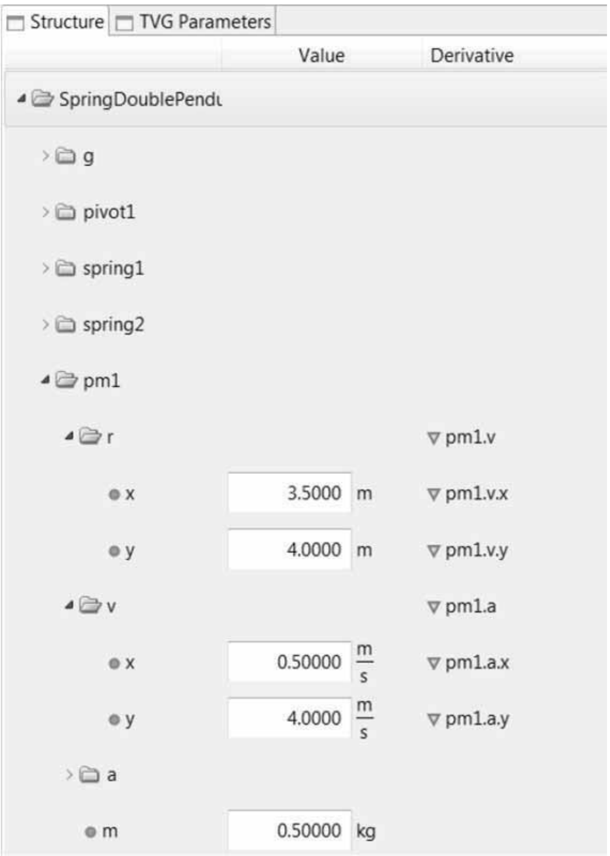
In the first the basic variables are declared and they are initialized with appropriate values. You do not really need to have any Java knowledge to understand, that this piece of code creates a vector g representing the earth acceleration, one pivot point, two springs and two point masses. In the second part (the constructor), these physical components are connected to each other. The first spring refers to the pivot and the first point mass, the first point mass refers to the first and the second spring, the second spring refers to both point masses and the second point mass refers to the second spring.

```
public class SpringDoublePendulum2D
     extends PhysicalSystem {
  public Vector2D g =
      new Vector2D(0, -9.81);
  public Vector2D pivot1 =
    new Vector2D(4, 8);
  public Spring2D spring1 =
    new Spring2D(25.5, 1, 1e5);
  public Spring2D spring2 =
    new Spring2D(25.5, 1, 1e5);
  public PointMass2D pm1 =
    new PointMass2D(3.5, 4, 0.5, 1, 0.5, g);
  public PointMass2D pm2 =
    new PointMass2D(5.5, 1.5, 0.2, 0,0.5, g);
  public SpringDoublePendulum2D() {
    spring1.r1 = pivot1;
    spring1.r2 = pm1.r;
    spring2.r1 = pm1.r;
    spring2.r2 = pm2.r;
    pm1.springs =
      new Spring2D[] { spring1, spring2 };
    pm2.springs = new Spring2D[] { spring2 };
  }
}
```

This piece of program code represents a physical system that is ready to be loaded and run. Be aware, that this physical system does not contain any formula. All you have to do is create such components and connect them with one another. The physical formulas are inside these components. This is why a spring 'knows', how to calculate its force as soon as it is connected with two endpoints. If a point mass is connected to one or several springs, then the point mass 'knows', that the forces from the springs have to be applied to the point mass.

Exercises from this category are similar to category 1. The students should learn about specific physical phenomena. Other than in category 1, the physical system is not ready made, but the students can compose them by themselves.

Thereby, they can also vary the model and build their own physical model. Example: Build a chain of point masses interconnected with springs, stimulate the first point mass and see how a wave moves through the physical system.



Structure | TVG Parameters

	Value		Derivative
SpringDoublePendu			
g			
pivot1			
spring1			
spring2			
pm1			
r			▽ pm1.v
● x	3.5000	m	▽ pm1.v.x
● y	4.0000	m	▽ pm1.v.y
v			▽ pm1.a
● x	0.50000	$\frac{m}{s}$	▽ pm1.a.x
● y	4.0000	$\frac{m}{s}$	▽ pm1.a.y
a			
● m	0.50000	kg	

Figure 3: Structure of spring mass double pendulum.

Basically, this example uses object oriented programming techniques. The students, however, do not necessarily have to understand the underlying concepts. The program code is used as a specific kind of notation for describing physical components and the relationships between these components. This is how the students get used to object oriented modeling in an application oriented fashion – without yet knowing the underlying object oriented concepts such as classes, instances, constructors, and inheritance. After loading the physical system, the structure of this physical system with its hierarchy of components and subcomponents and its derivation relationships is visually represented in the Physolator framework (see Figure 3).

For such systems, one may also provide graphics components that automatically visually represent the state of the system on the screen. *MechanicsTVG* is a generic 2D graphics component for visualizing simple mechanical systems. Adding the following program code to the above physical system results in an additional graphics component drawing the point masses and springs on the screen (see figure 4).

```
public void initGraphicsComponents(
  GraphicsComponents g, Structure s,
  Recorder r, SimulationParameters sp) {
    g.addTVG(new MechanicsTVG(this, s, r));
}
```

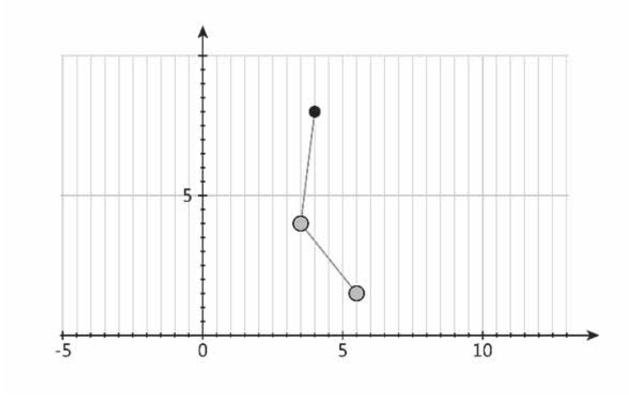


Figure 4: Spring mass double pendulum.

2.4 Category 4: Graphics programming

Simple two dimensional graphics programming is well suited for programming exercises at the beginner's level. When the program draws lines and circles on the screen, students get an immediate visual feedback, showing if the program is doing what it is supposed to do. At the same time, the students also have to deal with geometry in an applied manner.

For a simple two dimensional graphics programming, you would not necessarily need the Physolator framework. If your students, however, have already built a physical model such as the examples from category 2 or 3, then it makes perfect sense to define an exercise, where the students program a graphics component that graphically displays the state of the physical system.

During a physical simulation, the graphics are in motion. The graphic represents the physical system state and the physical system state changes during time. Time dependent graphics are nothing but movies.

With the Physolator, you can also produce movies without having to deal with any physics: Build an empty physical system, let this physical system load a graphics component and from inside the graphics component access the actual value of the simulation time *t*.

The Physolator framework also supports three dimensional graphics based on OpenGL. Three dimensional graphics programming is far more challenging. In such an exercise the students have to learn, how to define a 3D environment with a given camera position and camera direction, certain sources of light, 3D objects of certain shapes and a certain reflection behavior of their surfaces, fog, etc..

2.5 Category 5: Object oriented programming

Teaching object oriented programming concepts is not easy. The teaching goals of such lectures are:

- Learn the basic language constructs and concepts of an object oriented programming language: classes, instance, constructors, encapsulation and inheritance.
- Learn, how to apply these language constructs for developing complex software system and build a software structure that is designed for reusability.

Object oriented programming pays out when developing complex software. Unfortunately, in a programming class the time for the practical exercises is very limited and this is why in such exercises usually only small pieces of program code are produced. For small sized problems, it is hard to explain, that object oriented techniques are superior to the quick and dirty approach without a welldefined object oriented structure.

In a category 5 exercise, students shall use object oriented programming techniques to build their own physical components and use them within physical systems. Before starting with a category 5 exercise, students should first do some category 3 exercises. In a category 3 exercise, students have learn, how to build a physical system by composing given physical components. In a category 3 exercise, the students are using the notations from object oriented programming without necessarily understanding, that this program code is about object oriented programming and that the program code deals with classes, uses constructors and creates instances of classes.

As a preparation for a category 5 exercise, students have to learn the meaning of these language concepts. Then the students shall use these concepts to build their own physical components and use them inside physical systems.

An exercise from this category could ask the students to build the physical components from scratch. As an example, the program code for point masses and springs is easy to implement. Examples with point masses and springs can be found in [4]. This book also thoroughly discusses different kinds of modeling techniques using these examples. Other examples for physical components and the object oriented approaches being used to implement them, can be found in [3] and [5].

An object oriented modeling exercise does not necessarily have to start from scratch. Inheritance can also start with a given example. Sample-Scenario: Build a physical system which is based on the following physical system, but replace the graphics component with your own graphics component and use different simulation parameters. Another example, where inheritance is used on the level of physical components: Build a physical system, that is based on the string double pendulum example from category 3, but replace the given linear springs (Hooke's law) with nonlinear plate springs. The following program code uses inheritance to define *PlateString2D* a son class of *Spring2D*. Due to the fact, that *PlateString2D* inherits from *Spring2D*, one can modify the spring double pendulum code by replacing all occurrences of *Spring2D* by *PlateString2D* and then run the same example with plate springs.

The program code below adds the relevant parameters of a plate springs (*F0*, *p* and *h*) and uses overwriting to provide the class with a new implementation of method computeF in order to define the physical behavior of the plate spring.

```
public class PlateSpring2D
                     extends Spring2D {
  public double F0;
  public double p;
  public double h;
  public PlateSpring2D(double F0, double p,
        double h) {
    this.F0 = F0;
    this.p = p;
    this.h = h;
  }
  public double computeF(double distance) {
    double sigma = distance / h;
    return F0 * sigma *
      ((1 - sigma) * (1 - 0.5 * sigma) *
      Math.pow(h / p, 2) + 1);
  }
}
```

2.6 Category 6: Learning basics about physical simulation

In the exercises from the previous categories, students run physical simulations without necessarily understanding, how physical simulations are executed. There are quite some facts, that students could learn about physical simulations and there are exercises for deepening the understanding. This kind of knowledge is not only applicable to physical simulation, but also gives the students a better understanding about the core principles of computer games and computer based animations.

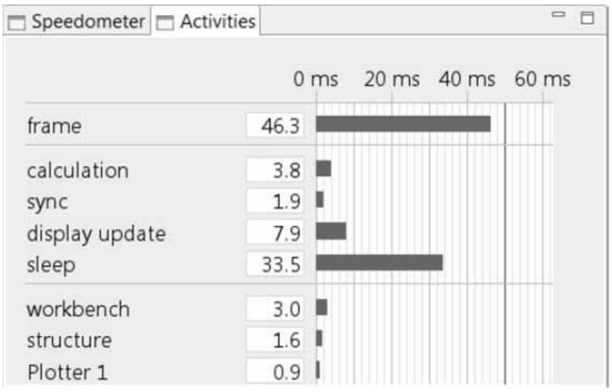
The teaching goals to be pursued in this category of exercises.

- Understand, that a physical simulation is executed in a time discrete manner.
- Learn, that a numerical simulation has limited accuracy. Understand, that a smaller step width results in a more precise simulation run, but result in a higher computational effort.
- Learn, that there some physical systems such as a simple trajectory, where there are algebraic ways to describe the behavior of a physical system with respect to time (closed solutions). In most other cases, the computer based simulations have to be used.
- Learn about different kinds of numerical procedures: fixed step width vs. flexible step width, single step vs. multi-step procedures.

The exercises from this category shall deepen the understanding about this domain. The Physolator framework provides several simulation parameters. The simulation parameters allow the user to choose, how the simulation is carried out. Among others, the user may choose a step width and the user may also choose among different kinds of numerical procedures (ODE solvers) such as Runge-Kutta, Adam-Bashforth, Cash-Karp and Dormand-Prince. In an exercise, students could be asked to load a given physical system and then find the right simulation parameter settings so the simulation runs with a highest possible accuracy and with a minimum of computational effort (CPU time consumption).

The Moon-Earth-Satellite from figure 1 would be well suited for this purpose – but one could also use any other physical system. In this exercise, students have to choose the right numerical procedure and the right step width. The computational effort for every simulation run can be monitored using the Physolator's built in monitoring tools (see Figure 5).

Students working on this exercise will quickly realize, that the accuracy and the computational effort are depending on the underlying numerical procedure as well as the step width. They will first have to produce a very precise result using very small step widths thus decreasing the truncation error. Too small step widths, however, lead to an increase of the round-off error. By working on such an exercise, the student get an awareness of the fact, that simulations are run in a time discrete manner and that the total error increases the longer the simulation runs. In a follow-up lecture one could explain the underlying theoretical problems.



frame	46.3
calculation	3.8
sync	1.9
display update	7.9
sleep	33.5
workbench	3.0
structure	1.6
Plotter 1	0.9

Figure 5: Performance monitor.

A variation of this kind of exercise would work with a physical system, where a direct, algebraic solution exists. Examples for such physical systems: simple trajectory, damped point-spring-pendulum. Other than in the previous type of exercise, a precise solution is given and therefore it is easy to compute the error at any time. Working with such an example would also deepen the awareness, that at least for some physical systems, the behavior of physical systems can be described using closed equations and a physical simulation is not necessarily required in these cases.

3 Summary and Conclusion

This paper has presented different categories of exercises related to physical simulation and it has been explained, how the Physolator simulation framework can be used as an infrastructure for such exercises.

It has been shown, that the different types of exercises pursue different kinds of teaching goals. All of the teaching goals for such exercises are in the STEM field. In many exercises, several STEM qualifications are needed: physics, programming, mathematics. These qualifications have to be combined when working on the exercise. This is why physical simulation is a domain, where students not only acquire knowledge from different STEM fields, but also learn, how to work in an interdisciplinary manner and combine these skills.

References

[1] Eisenbiegler D. “The Software Architecture of the Physolator – a Physical Simulation Framework”, MSAM 2015, Atlantis Press, pp. 61-64.

[2] Eisenbiegler D. “Object Oriented Modeling and Simulation with the Physolator – Getting Started”, https://opus.hs-furtwangen.de/frontdoor/index/index/docId/614, 2016.

[3] Eisenbiegler D. “A Generic Particle Modeling Library for Fluid Simulation”, AMSM 2016, Atlantis Press.

[4] Eisenbiegler D.“Objektorientierte Model-lierung und Simulation physikalischer Systeme mit dem Physolator”, BoD Norderstedt, 2015.

[5] Eisenbiegler D. “An Object Oriented Library for Acoustics Simulation Based on the Physolator Simulation Framework”, CMSAM 2016, DEStech Publications.

[6] Bloom B. “Taxonomy of Educational Objectives”, Pearson Eduction, 1984

[7] Pintó R, Couso D (editors). “Contributions from Science Education Research”, Springer, 2014

[8] Papadouris N, Hadjigeorgiou A, Constantinou C (editors). “Insights from Research in Science Teaching and Learning”, Springer, 2013.

[9] Mikelskis-Seifert S, et.al.. “Physik-Methodik”, Cornelson Stuttgart, 2010.

A Norm-based Comparison of Approaches in ARGESIM Benchmark C17R 'Modelling and Simulation of a SIR-type Epidemic with Cellular Automata and Ordinary Differential Equations' using MATLAB

Stephan Reichl[1*], Florian Miksch[2], Günter Schneckenreither[1,2]

[1]Institute of Analysis and Scientific Computing, TU Wien, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria; [*]*reichl.stephan@gmail.com*

[2]dwh simulation services, Neustiftgasse 57-59, 1070 Vienna, Austria

SNE 27(1), 2017, 53-59, DOI: 10.11128/sne.27.bn17.10368
Received: May 10, 2016, Revised: December 20, 2017
Accepted: January 30, 2017
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. This model comparison quantifies the difference between the results of a population model, using ordinary differential equations, and an inhomogeneous spatial approach, using a lattice gas cellular automaton. Both approaches describe a classical SIR-epidemic. The system definition of the epidemic, model specification and parameter identification were already formulated by Miksch et al [1]. For the sake of quantifiable comparisons a norm was introduced. The tasks from the revised ARGESIM Comparison 17 [1] were completed and the results were analysed and visualised. Every result was interpreted according to the knowledge of the modelling approaches and the setup of the experiment at hand.

Introduction

This model comparison follows the outlined tasks of the revised ARGESIM Comparison 17 [1] and is targeted on quantifying the difference of SIR-type epidemic simulations, which were either modelled by classical ordinary differential equations (ODEs) or a lattice gas cellular automaton (LGCA). Both modelling approaches use the same system definition and therefore the same basis for parameter calculation. From now on we will use the term "population group" to describe any of the following cohorts: susceptible, infected or recovered individuals.

Although both models are strictly defined in the revised Comparison 17[1] some changes or specifications had to be made:

- **Periodic boundary conditions** were used in the LGCA as follows: when an individual passes through any boundary of the lattice, it reappears on the opposite side of the lattice at the same position inside the cell and therefore with the same movement direction as before.
- The presented formula for the **calculation of the dimension** n of the LGCA's lattice yields a number, which is too small for certain populations where the contact rate C is chosen to be 5. This should be the maximum in an LGCA with hexagonal cells. The solution to this problem is to simply round up instead of down, as it is shown in the following.
$$n = \left\lceil \sqrt{\frac{5\,(N-1)+C}{6C}} \right\rceil \tag{1}$$
- The specification of **hard interventions with the LGCA model** was extended by the following step: Since the LGCA approach operates on discrete values the fraction of the affected population group had to be rounded down after the multiplication with the factor f_H.
- In the **ODE model** we were confronted with a **limitation** on the variation spectrum of the probability rates α and β. If we choose the value 1 for either of them then we get an numerically undefined term ($\log(0)$) when we try to carry out the parameter identification as instructed.

1 Implementation

The simulations were performed using a MacBook Air (11inch, End 2010) running Mac OSX 10.11.6 and the Software MATLAB R2015b. It should be pointed out that the results of the LGCA vary in every iteration, because of the usage of probability principles. The realization of the ODE model always yields exactly the same results (provided that the same ODE-solver is used). To ensure exact reproducibility a large number of iterations with the LGCA model would have been necessary. Then the average values of the infected individuals at every time step would have been used in the comparisons, but this would have gone beyond the scope of this paper. Here we used one round of simulations, after a lot of test runs, to generate the results.

1.1 Differential Equations Model

The system of differential equations was solved using the MATLAB ODE-solver ode45, which is used to solve systems of nonstiff differential equations. Furthermore we only calculated the solutions for integer time steps starting with 1, to ensure comparability between the continuous ODE-model and the discrete LGCA-model. The solution contained the number of individuals for each population group at every integer time step.

1.2 Cellular Automaton

The model was implemented as described in the revised comparison [1] with one exception, which was previously explained resulting in equation 1 for the dimension of the lattice.

The initial conditions, represented by the individuals of every population group at time $t = 0$, were uniformly distributed in the cells of the lattice, which was implemented as a matrix $M \in \mathbb{N}^{n \times n \times 6}$. This was realized by randomly permuting a vector, with the length equal to the number of free cells ($6n^2$), filled with the integers from 1 to $6n^2$. Then it was possible to take the first x values of that vector as indexes for the individuals of one population group, where x is its initial condition.

After initializing the LGCA the movement phase is implemented according to the rules, which were defined in the revised comparison [1] and extended by the boundary condition presented in the Introduction.

Then the collision phase is realized by using the FHP-I collision rules. Followed by that, the infection- and recovery-phase is implemented. In this phase we simulate the contacts between individuals in the same hexagonal-cell. With every contact between an infected individual and a susceptible one, there is a certain possibility of an infection. Additionally for an infected individual there is the possibility to recover in every single time step. In order to assert synchronicity, the infection and recovery takes place in a buffered copy of the whole lattice.

Functions of probability are used twice in the implementation: when we determine the distribution of the initial conditions on the lattice (this happens only once per simulation) and every time when there is the possibility of an infection or recovery of an individual.

At the end of every iteration/time step, we sum up all the population groups and save them in association with the according point in time.

1.3 Results

The result of an experiment always consists of a difference-determining value, which will be introduced in the next section, and five plots: each modelling approach with the corresponding SIR-curves, one which showed the infected curves of both simulations and the 1-Norm and 2-Norm of the difference in infected individuals per time step. For every experiment or variation of parameters a MATLAB-file was generated and the simulations took place with the help of a script which loaded the parameter file and generated the result document automatically for every experiment. As a result of this procedure it was ensured that in every experiment both approaches used the same base-parameters.

2 Comparison Norm

It is always useful to have one value which determines the difference between two functions. We decided that the curve of infected individuals best describes an epidemic over the course of time. For further analysis and quantifiable comparability we introduced a value to determine the exact difference between the two curves, because it often happens that the curves are very similar but offset by a certain time.

We came to the conclusion that the best way to quantify and ensure comparability between experiments is to determine the area of the absolute difference between both curves of infected individuals divided by the total population (N). This can formally be written as

$$\frac{\int_0^T |y(t)_{ode} - y(t)_{ca}|\, dt}{N} \tag{2}$$

Since the comparison takes place under discrete circumstances with equally spaced time steps we have to simplify equation 2 with the help of the trapezoid rule to

$$\frac{\sum_{t=0}^{T} \Big(|y(t)_{ode} - y(t)_{ca}| + |y(t+1)_{ode} - y(t+1)_{ca}| \Big)}{2N} \tag{3}$$

In MATLAB this was implemented by calculating the absolute difference at every discrete point in time (1-Norm) and the use of the function `trapz()`.

3 Parameters & Figures

The parameters of Table 1 were used to conduct different experiments by varying only one at a time.

S_0	I_0	C	α	β	T_{end}
9500	500	4	0.1	0.1	100

Table 1: Default configuration of the available parameters.

The population group Recovered was always initialized with zero individuals ($R_0 = 0$) and the total population (N) was set constant with 10000 individuals.
With this data pool as setup, both model approaches were simulated and compared. The results are different plots and a value for the comparison of the difference between the two approaches in different experiments, as described before. In Figure 1, 2 and 3 we present the different kinds of plots that were used for every experiment with the parameters of Table 1. The calculation for the comparison norm for these parameters results in

$$d(ODE, LGCA) = 1.4844 \tag{4}$$

In further consequence we will not display these three plots for every parameter configuration because of lack of room, readability and comparability. Therefore bar charts will be used with the help of the comparison norm, which was introduced earlier in equation 3.

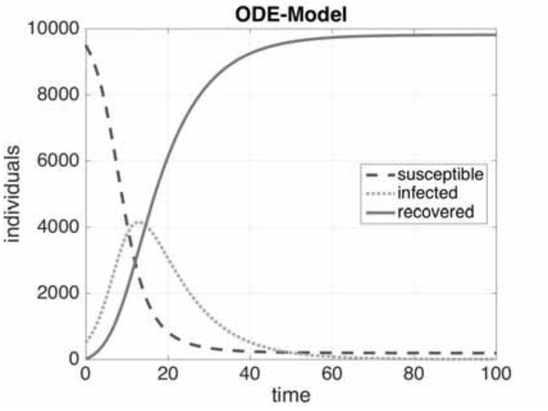


Figure 1: The epidemic over time with the ODE-model.

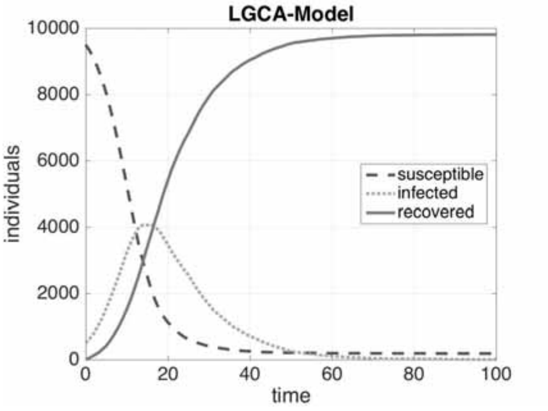


Figure 2: The epidemic over time with the LGCA-model.

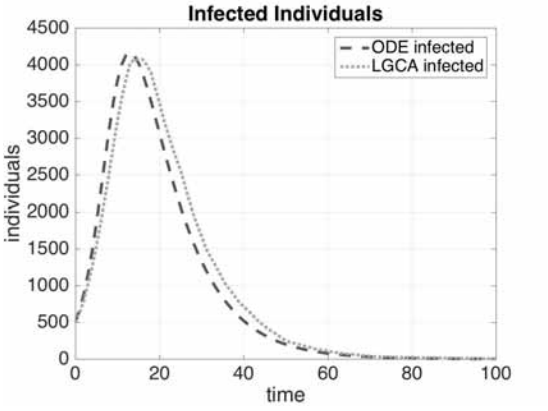


Figure 3: The Infected individuals over time for each model.

In the following we will always present the used parameters, which deviate from the default values from Table 1, as tables and the results as bar plots.

4 Task 1) Parameter Variation

In the first part of task 1 simulations with different variations of one parameter at a time were performed. The values of the following parameters were varied: initial number of infected individuals I_0 (and therefore S_0), the average of contacts per time step C, the infection probability α and the recovery probability β. The different variations of every parameter are displayed in Table 2, 3, 4 and 5 respectively. In the second part only the contact rate C was varied while the product of $C\alpha$ had to remain constant as it can be seen in Table 6.

4.1 Task 1) a) - Variation of I_0

	1	**2**	**3**	**4**	**5**	**6**	**7**
I_0	5	50	250	500	750	1000	5000

Table 2: Parameter variation of I_0.

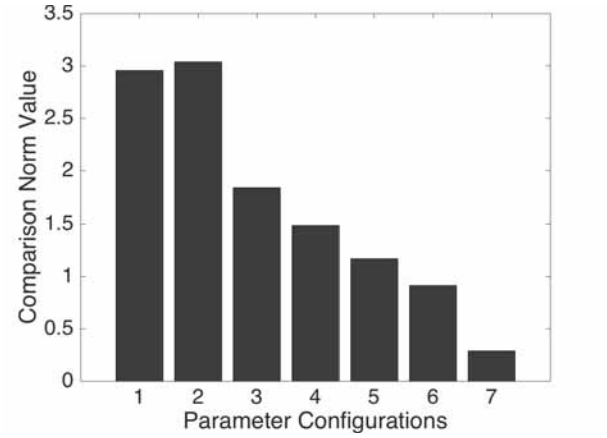


Figure 4: Results of the variation of I_0.

4.2 Task 1) a) - Variation of C

C	1	2	3	4	5

Table 3: Parameter variation of C.

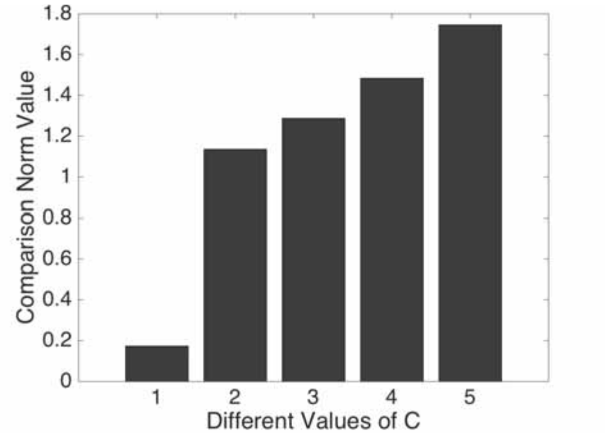


Figure 5: Results of the variation of C.

4.3 Task 1) a) - Variation of α

	1	**2**	**3**	**4**	**5**	**6**	**7**
$\alpha \times 10^2$	1	5	7.5	10	20	50	99.99

Table 4: Parameter variation of α.

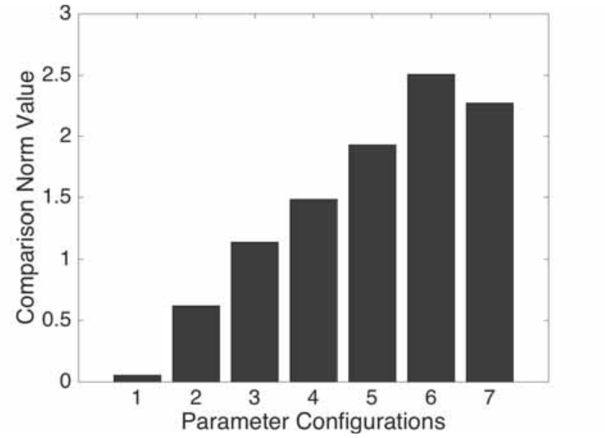


Figure 6: Results of the variation of α.

4.4 Task 1) a) - Variation of β

	1	2	3	4	5	6	7
β	0.001	0.01	0.05	0.1	0.2	0.25	0.5

Table 5: Parameter variation of β.

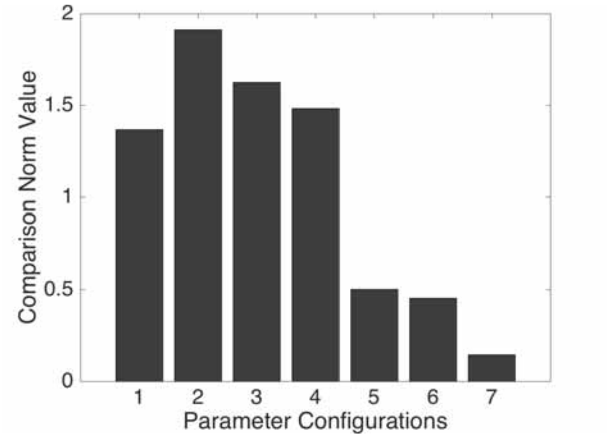


Figure 7: Results of the variation of β.

4.5 Task 1) a) - Interpretation

Looking at the results, the following hypotheses can be formulated

- We observe in Figure 4 that more infected individuals at the start result in smaller differences.
- Figure 5 indicates that the more contacts happen, the greater the difference in the results get.
- We can see in Figure 6 that greater infection probability results in larger norm values.
- The opposite seems to be true for the recovery probability as illustrated in Figure 7. Greater recovery probability results in smaller norm values.

The first and last point can probably be explained by the fact that the configurations, which yield small norm values, compensate for the spatial component in the LGCA model. Furthermore the recovery phase does not take into account where a individual is positioned and therefore the spatial component of the LGCA has no impact on it. The second and third hypothesis may arise from the fact that the movement of individuals is modelled differently in the two approaches.

4.6 Task 1) b) - Variation of C with $C\alpha = const$

	1	2	3	4	5
C	1	2	3	4	5
α	0.4	0.2	$0.1\dot{3}33$	0.1	0.08

Table 6: Parameter variation of C and α with $C\alpha = const$.

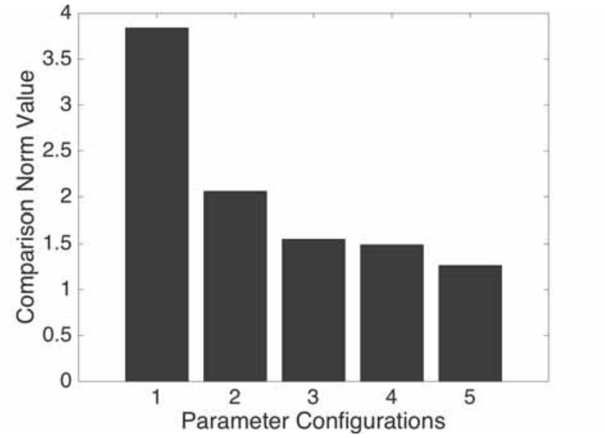


Figure 8: Results of the variation of C and α with $C\alpha = const$.

4.7 Task 1) b) - Interpretation

In Figure 8 it can be observed that for $C \geq 2$ the norm values drop below 2 but never 1. We want to point out that if we compare Figure 5 and Figure 8 and the tables, respectively we have to come to the conclusion that the adapted infection probability α is not causing any improvement compared to the situation with constant α. This should not be surprising as we have already seen in Figure 6 that the norm increases if α is increased.

5 Task 2) Interventions

The interventions were implemented with 60 time steps, because they cause an earlier end of an epidemic. In the **Soft Interventions** we modelled the decrease (within $\Delta t = 10$) of the intervened parameter (α) with a smooth step, instead of a linear one, because it is more realistic and the difference was smaller than 0.1. As **Hard Intervention** strategy we chose to vaccinate susceptible individuals ($f_H S$). Both interventions start when a certain threshold ($f_T N = I$) is reached. We used the default configuration from Table 1 and only varied the intervention specific parameters as stated in Table 7 and 8.

5.1 Task 2) a) - Soft Intervention

	1	2	3	4	5
f_T	0.1	0.2	0.3	0.2	0.2
f_S	0.5	0.5	0.5	0.1	0.25

Table 7: Soft Intervention parameter variation of f_T and f_S.

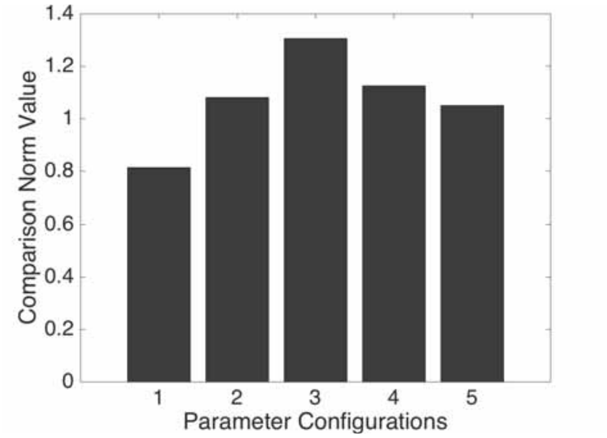


Figure 9: Soft Intervention parameter variation of f_T and f_S.

5.2 Task 2) b) - Hard Intervention

	1	2	3	4	5
f_T	0.1	0.2	0.3	0.2	0.2
f_H	0.1	0.1	0.1	0.25	0.5

Table 8: Hard Intervention parameter variation of f_T and f_H.

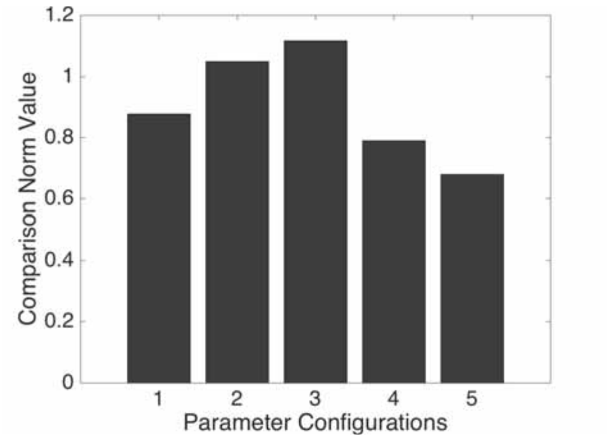


Figure 10: Hard Intervention parameter variation of f_T and f_H.

5.3 Task 2) Interpretation

Looking at Figure 9 and Figure 10 we can see that the norm values of all the experiments with both kinds of interventions always lie between 0.6 and 1.4. Therefore it is safe to say that by varying the intervention parameters no parameter regions could be detected where the models do not behave very similar. Often the difference between the results was caused by the lacking behind of the LGCA approach because after reaching the defined threshold ($f_T N = I$), which triggers the intervention, actions took place in the next time step and not at that moment, as it is realised in the ODE model.

6 Task 3) Spatial Inhomogeneity

We replaced the implemented movement rules of the LGCA with random movements to achieve a constant homogeneous mixture of the population. After every time step the individuals were randomly distributed on the lattice. We chose four bad and four good results of previous experiments and their parameters (as shown in Table 9), to see if the new rule changes the outcome. The bold values in table 9 indicate deviations from Table 1.

	S_0	I_0	C	α	β	T_{end}
1	9500	500	4	**0.9999**	0.1	100
2	9500	500	**1**	**0.4**	0.1	100
3	**9950**	**50**	4	0.1	0.1	100
4	9500	500	**2**	**0.2**	0.1	100
5	9500	500	**1**	0.1	0.1	100
6	9500	500	4	**0.01**	0.1	100
7	9500	500	4	0.1	**0.5**	100
8	**5000**	**5000**	4	0.1	0.1	100

Table 9: Parameters for Spatial Inhomogeneity Comparison.

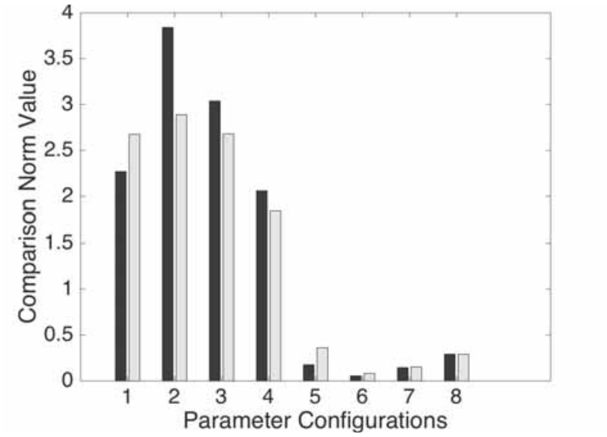


Figure 11: Results with (blue) and without (yellow) Spatial Inhomogeneity.

6.1 Task 3) - Interpretation

As we can see, in Figure 11, nearly every scenario produced better results with the random movement. Only configuration 1 and 5 experienced a worsening in the result. These are probably outlier. In the results of configuration 6,7 and 8, where the results were already good, no significant change was noticed.

References

[1] Miksch F, Haim C, Schneckenreither G, Popper N. Modelling and Simulation of a SIR-type Epidemic with Cellular Automata and Ordinary Differential Equations - Definition ARGESIM Benchmark C17R. *Simulation Notes Europe*. 2015;25(1):49–54.

EUROSIM 2019

9th EUROSIM Congress on Modelling and Simulation

La Rioja, Logroño, Spain, July 2019

EUROSIM Congresses are the most important modelling and simulation events in Europe. For EUROSIM 2019, we are soliciting original submissions describing novel research and developments in the following (and related) areas of interest: Continuous, discrete (event) and hybrid modelling, simulation, identification and optimization approaches. Two basic contribution motivations are expected: M&S Methods and Technologies and M&S Applications. Contributions from both technical and non-technical areas are welcome.

Congress Topics The EUROSIM 2019 Congress will include invited talks, parallel, special and poster sessions, exhibition and versatile technical and social tours. The Congress topics of interest include, but are not limited to:

Intelligent Systems and Applications
Hybrid and Soft Computing
Data & Semantic Mining
Neural Networks, Fuzzy Systems & Evolutionary Computation
Image, Speech & Signal Processing
Systems Intelligence and Intelligence Systems
Autonomous Systems
Energy and Power Systems
Mining and Metal Industry
Forest Industry
Buildings and Construction
Communication Systems
Circuits, Sensors and Devices
Security Modelling and Simulation
Bioinformatics, Medicine, Pharmacy and Bioengineering
Water and Wastewater Treatment, Sludge Management and Biogas Production
Condition monitoring, Mechatronics and maintenance
Automotive applications
e-Science and e-Systems
Industry, Business, Management, Human Factors and Social Issues
Virtual Reality, Visualization, Computer Art and Games
Internet Modelling, Semantic Web and Ontologies
Computational Finance & Economics
Simulation Methodologies and Tools
Parallel and Distributed Architectures and Systems
Operations Research
Discrete Event Systems
Manufacturing and Workflows
Adaptive Dynamic Programming and Reinforcement Learning
Mobile/Ad hoc wireless networks, mobicast, sensor placement, target tracking
Control of Intelligent Systems
Robotics, Cybernetics, Control Engineering, & Manufacturing
Transport, Logistics, Harbour, Shipping and Marine Simulation

Congress Venue / Social Events The Congress will be held in the City of Logroño, Capital of La Rioja, Northern Spain. The main venue and the exhibition site is the University of La Rioja (UR), located on a modern campus in Logroño, capital of La Rioja, where 7500 students are registered. The UR is the only University in this small, quiet region in Northern Spain. La Rioja is where the Monasteries of San Millán de la Cogolla, cradle of the first words written in the Spanish language, are situated, sites included in UNESCO's World Heritage List in 1996. Of course, social events will reflect this heritage – and the famous wines in la Rioja.

CongressTeam: The Congress is organised by CAE CAE-SMSG, the Spanish simulation society, and Universidad de la Rioja.

Info: Emilio Jiménez, EUROSIM President, emilio.jimenez@unirioja.es
Juan Ignacio Latorre, juanignacio.latorre@unavarra.es

www.eurosim.info

SNE Simulation News

EUROSIM Data and Quick Info

EUROSIM 2019

10th EUROSIM Congress on Modelling and Simulation

La Rioja, Logroño, Spain, July 2019

Contents

Simulation Notes Europe SNE is the official membership journal of EUROSIM and distributed / available to members of the EUROSIM Societies as part of the membership benefits.
If you have any information, announcement, etc. you want to see published, please contact a member of the editorial board in your country or the editorial office. For scientific publications, please contact the EiC.

This *EUROSIM Data & Quick Info* compiles data from **EUROSIM** societies and groups: addresses, weblinks, and officers of societies with function and email, to be published regularly in **SNE** issues. This information is also published at EUROSIM's website *www.eurosim.info*.

SNE Reports Editorial Board

EUROSIM Emilio Jiménez, *emilio.jimenez@unirioja.es*
Andreas Körner, *andreas.koerner@tuwien.ac.at*
Miguel Mujica Mota, *m.mujica.mota@hva.nl*
ASIM A. Körner, *andreas.koerner@tuwien.ac.at*
CEA-SMSG Emilio Jiminez, *emilio.jimenez@unirioja.es*
CROSSIM Vesna Dušak, *vdusak@foi.hr*
CSSS Mikuláš Alexík, *alexik@frtk.utc.sk*
DBSS M. Mujica Mota, *m.mujica.mota@hva.nl*
FRANCOSIM Karim Djouani, *djouani@u-pec.fr*
HSS András Jávor, *javor@eik.bme.hu*
ISCS M. Savastano, *mario.savastano@unina.it*
LIOPHANT F. Longo, *f.longo@unical.it*
LSS Yuri Merkuryev, *merkur@itl.rtu.lv*
PSCS Zenon Sosnowski, *zenon@ii.pb.bialystok.pl*
RNSS Y. Senichenkov, *senyb@dcn.icc.spbstu.ru*
SIMS Esko Juuso, *esko.juuso@oulu.fi*
SLOSIM Vito Logar, *vito.logar@fe.uni-lj.si*
UKSIM A. Orsoni, *A.Orsoni@kingston.ac.uk*
KA-SIM Edmond Hajrizi, *info@ka-sim.com*
MIMOS Paolo Proietti, *roma@mimos.it*
ROMSIM Marius Radulescu, *mradulescu@ici.ro*
Albanian Society Kozeta Sevrani, *kozeta.sevrani@unitir.edu.al*

SNE Editorial Office /ARGESIM

→ *www.sne-journal.org, www.eurosim.info*
office@sne-journal.org, Andreas Körner, (info, news)
eic@sne-journal.org, Felix Breitenecker (publications)
✉ SNE Editorial Office, Andreas Körner c/o ARGESIM / Mathematical Modelling & Simulation Group, TU Wien /101,
Wiedner Haupstrasse 8-10, 1040 Vienna , Austria

EUROSIM Federation of European Simulation Societies

General Information. **EUROSIM**, the Federation of European Simulation Societies, was set up in 1989. The purpose of **EUROSIM** is to provide a European forum for simulation societies and groups to promote advancement of modelling and simulation in industry, research, and development. → *www.eurosim.info*

Member Societies. **EUROSIM** members may be national simulation societies and regional or international societies and groups dealing with modelling and simulation. At present **EUROSIM** has 16 *Full Members* and 2 *Observer Members*, and one member candidate.

ASIM	Arbeitsgemeinschaft Simulation *Austria, Germany, Switzerland*
CEA-SMSG	Spanish Modelling and Simulation Group *Spain*
CROSSIM	Croatian Society for Simulation Modeling *Croatia*
CSSS	Czech and Slovak Simulation Society *Czech Republic, Slovak Republic*
DBSS	Dutch Benelux Simulation Society *Belgium, Netherlands*
FRANCOSIM	Société Francophone de Simulation *Belgium, France*
HSS	Hungarian Simulation Society; *Hungary*
ISCS	Italian Society for Computer Simulation *Italy*
KA-SIM	Kosovo Simulation Society, *Kosovo*
LIOPHANT	LIOPHANT Simulation Club *Italy & International*
LSS	Latvian Simulation Society; *Latvia*
PSCS	Polish Society for Computer Simulation *Poland*
MIMOS	Italian Modelling and Simulation Association, *Italy, Observer Member*
RNSS	Russian National Simulation Society *Russian Federation*
ROMSIM	Romanian Society for Modelling and Simulation, *Romania, Observer Member*
SIMS	Simulation Society of Scandinavia *Denmark, Finland, Norway, Sweden*
SLOSIM	Slovenian Simulation Society *Slovenia*
UKSIM	United Kingdom Simulation Society *UK, Ireland*

EUROSIM Board / Officers. EUROSIM is governed by a board consisting of one representative of each member society, president and past president, and representatives for **SNE** Simulation Notes Europe. The President is nominated by the society organising the next **EUROSIM** Congress. Secretary, Secretary to the Board, and Treasurer are elected out of members of the board.

President	Emilio Jiménez (CAE-SMSG), *emilio.jimenez@unirioja.es*
Past President	Esko Juuso (SIMS) *esko.juuso@oulu.fi*
Secretary	M. Mujica Mota (DBSS), *m.mujica.mota@hva.nl*
Treasurer	Felix Breitenecker (ASIM) *felix.breitenecker@tuwien.ac.at*
Secretary to the Board	Andreas Körner *Andreas.koerner@tuwien.ac.at*
SNE Representative	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*

SNE – Simulation Notes Europe. **SNE** is a scientific journal with reviewed contributions as well as a membership newsletter for **EUROSIM** with information from the societies in the *News Section*. **EUROSIM** societies are offered to distribute to their members the journal **SNE** as official membership journal. **SNE** Publishers are **EUROSIM**, ARGESIM and ASIM.

SNE Editor-in-Chief	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*

→ *www.sne-journal.org,*
✉ *office@sne-journal.org*

EUROSIM Congress. **EUROSIM** is running the triennial conference series **EUROSIM** Congress. The congress is organised by one of the **EUROSIM** societies.

EUROSIM 2019, the 10th EUROSIM Congress, will be organised by CAE-SMSG, the Spanish simulation society, in La Rioja, Logroño, Spain, in July 2019.

Chairs / Team EUROSIM 2019
Emilio Jiménez, EUROSIM President, *emilio.jimenez@unirioja.es*
Juan Ignacio Latorre, *juanignacio.latorre@unavarra.es*

→ *www.eurosim.info*

EUROSIM Member Societies

ASIM
German Simulation Society
Arbeitsgemeinschaft Simulation

ASIM (Arbeitsgemeinschaft Simulation) is the association for simulation in the German speaking area, servicing mainly Germany, Switzerland and Austria. ASIM was founded in 1981 and has now about 600 individual members, and 90 institutional or industrial members.

→ *www.asim-gi.org* with members' area

✉ *info@asim-gi.org, admin@asim-gi.org*

⊠ ASIM – Inst. f. Analysis and Scientific Computing
Vienna University of Technology
Wiedner Hauptstraße 8-10, 1040 Vienna, Austria

ASIM Officers	
President	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*
Vice presidents	Sigrid Wenzel, *s.wenzel@uni-kassel.de*
	T. Pawletta, *pawel@mb.hs-wismar.de*
Secretary	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
	A. Körner, *andreas.koerner@tuwien.ac.at*
Treasurer	Anna Mathe, *anna.mathe@tuwien.ac.at*
Membership Affairs	S. Wenzel, *s.wenzel@uni-kassel.de*
	W. Maurer, *werner.maurer@zhwin.ch*
	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
	F. Breitenecker, *felix.breitenecker@tuwien.ac.at*
Repr. EUROSIM	F. Breitenecker, *felix.breitenecker@tuwien.ac.at*
	A. Körner, *andreas.koerner@tuwien.ac.at*
Int. Affairs – GI Contact	N. Popper, *niki.popper@drahtwarenhandlung.at*
	O. Rose, *Oliver.Rose@tu-dresden.de*
Editorial Board SNE	T. Pawletta, *pawel@mb.hs-wismar.de*
	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
Web EUROSIM	A. Körner, *andreas.koerner@tuwien.ac.at*

Last data update December 2016

ASIM is organising / co-organising the following international conferences:

- ASIM Int. Conference 'Simulation in Production and Logistics' – bi-annual
- ASIM 'Symposium Simulation Technique' – bi-annual
- MATHMOD Int. Vienna Conference on Mathmatical Modelling – tri-annual

Furthermore, ASIM is co-sponsor of WSC – Winter Simulation Conference

ASIM Working Committee	
GMMS	Methods in Modelling and Simulation Th. Pawletta, *pawel@mb.hs-wismar.de*
SUG	Simulation in Environmental Systems J.Wittmann, *wittmann@informatik.uni-hamburg.de*
STS	Simulation of Technical Systems Walter Comerell, *Commerell@hs-ulm.de*
SPL	Simulation in Production and Logistics Sigrid Wenzel, *s.wenzel@uni-kassel.de*
EDU	Simulation in Education/Education in Simulation A. Körner, *andreas.koerner@tuwien.ac.at*
BIG DATA	Working Group Data-driven Simulation in Life Sciences; *niki.popper@drahtwarenhandlung.at*
	Working Groups for Simulation in Business Administration, in Traffic Systems, for Standardisation, etc.

CEA-SMSG – Spanish Modelling and Simulation Group

CEA is the Spanish Society on Automation and Control and it is the national member of IFAC (International Federation of Automatic Control) in Spain. Since 1968 CEA-IFAC looks after the development of the Automation in Spain, in its different issues: automatic control, robotics, SIMULATION, etc. In order to improve the efficiency and to deep into the different fields of Automation. The association is divided into national thematic groups, one of which is centered on Modeling, Simulation and Optimization, constituting the CEA Spanish Modeling and Simulation Group (CEA-SMSG). It looks after the development of the Modelling and Simulation (M&S) in Spain, working basically on all the issues concerning the use of M&S techniques as essential engineering tools for decision-making and optimization.

→ *http://www.ceautomatica.es/grupos/*

→ *emilio.jimenez@unirioja.es*
simulacion@cea-ifac.es

⊠ CEA-SMSG / Emilio Jiménez, Department of Electrical Engineering, University of La Rioja, San José de Calasanz 31, 26004 Logroño (La Rioja), SPAIN

CEA - SMSG Officers	
President	Emilio Jiménez, *emilio.jimenez@unirioja.es*
Vice president	Juan Ignacio Latorre *juanignacio.latorre@unavarra.es*
Repr. EUROSIM	Emilio Jiminez, *emilio.jimenez@unirioja.es*
Edit. Board SNE	Emilio Jiminez, *emilio.jimenez@unirioja.es*
Web EUROSIM	Mercedes Perez *mercedes.perez@unirioja.es*

Last data update June 2016

CROSSIM – Croatian Society for Simulation Modelling

CROSSIM-*Croatian Society for Simulation Modelling* was founded in 1992 as a non-profit society with the goal to promote knowledge and use of simulation methods and techniques and development of education. CROSSIM is a full member of **EUROSIM** since 1997.

→ *www.eurosim.info*

✉ *vdusak@foi.hr*

🖂 CROSSIM / Vesna Dušak
Faculty of Organization and
Informatics Varaždin, University of Zagreb
Pavlinska 2, HR-42000 Varaždin, Croatia

CROSSIM Officers	
President	Vesna Dušak, *vdusak@foi.hr*
Vice president	Jadranka Božikov, *jbozikov@snz.hr*
Secretary	Vesna Bosilj-Vukšić, *vbosilj@efzg.hr*
Executive board members	Vlatko Čerić, *vceric@efzg.hr*
	Tarzan Legović, *legovic@irb.hr*
Repr. EUROSIM	Jadranka Božikov, *jbozikov@snz.hr*
Edit. Board SNE	Vesna Dušak, *vdusak@foi.hr*
Web EUROSIM	Jadranka Bozikov, *jbozikov@snz.hr*

Last data update December2012

CSSS – Czech and Slovak Simulation Society

CSSS -The *Czech and Slovak Simulation Society* has about 150 members working in Czech and Slovak national scientific and technical societies (*Czech Society for Applied Cybernetics and Informatics, Slovak Society for Applied Cybernetics and Informatics*). The main objectives of the society are: development of education and training in the field of modelling and simulation, organising professional workshops and conferences, disseminating information about modelling and simulation activities in Europe. Since 1992, CSSS is full member of **EUROSIM**.

→ *www.fit.vutbr.cz/CSSS*

✉ snorek@fel.cvut.cz

🖂 CSSS / Miroslav Šnorek, CTU Prague
FEE, Dept. Computer Science and Engineering,
Karlovo nam. 13, 121 35 Praha 2, Czech Republic

CSSS Officers	
President	Miroslav Šnorek, *snorek@fel.cvut.cz*
Vice president	Mikuláš Alexík, *alexik@frtk.fri.utc.sk*
Scientific Secr.	A. Kavička, *Antonin.Kavicka@upce.cz*
Repr. EUROSIM	Miroslav Šnorek, *snorek@fel.cvut.cz*
Edit. Board SNE	Mikuláš Alexík, *alexik@frtk.fri.utc.sk*
Web EUROSIM	Petr Peringer, peringer@fit.vutbr.cz

Last data update December2012

DBSS – Dutch Benelux Simulation Society

The Dutch Benelux Simulation Society (DBSS) was founded in July 1986 in order to create an organisation of simulation professionals within the Dutch language area. DBSS has actively promoted creation of similar organisations in other language areas. DBSS is a member of **EUROSIM** and works in close cooperation with its members and with affiliated societies.

→ *www.DutchBSS.org*

✉ *a.w.heemink@its.tudelft.nl*

🖂 DBSS / A. W. Heemink
Delft University of Technology, ITS - twi,
Mekelweg 4, 2628 CD Delft, The Netherlands

DBSS Officers	
President	A. Heemink, *a.w.heemink@its.tudelft.nl*
Vice president	M. Mujica Mota, *m.mujica.mota@hva.nl*
Treasurer	M. Mujica Mota, *m.mujica.mota@hva.nl*
Secretary	P. M. Scala, p.m.scala@hva.nl
Repr. EUROSIM	M. Mujica Mota, *m.mujica.mota@hva.nl*
Edit. SNE/Web	M. Mujica Mota, *m.mujica.mota@hva.nl*

Last data update June 2016

FRANCOSIM - Société Francophone de Simulation

FRANCOSIM was founded in 1991 and aims to the promotion of simulation and research, in industry and academic fields.

✉ *djouani@u-pec.fr*

🖂 FRANCOSIM / Yskandar Hamam
Groupe ESIEE, Cité Descartes,
BP 99, 2 Bd. Blaise Pascal,
93162 Noisy le Grand CEDEX, France

FRANCOSIM Officers	
President	Karim Djouani, *djouani@u-pec.fr*
Treasurer	François Rocaries, *f.rocaries@esiee.fr*
Repr. EUROSIM	Karim Djouani, *djouani@u-pec.fr*
Edit. Board SNE	Karim Djouani, *djouani@u-pec.fr*

Last data update December2012

HSS – Hungarian Simulation Society

The Hungarian Member Society of EUROSIM was established in 1981 as an association promoting the exchange of information within the community of people involved in research, development, application and education of simulation in Hungary and also contributing to the enhancement of exchanging information between the Hungarian simulation community and the simulation communities abroad. HSS deals with the organization of lectures, exhibitions, demonstrations, and conferences.

→ *www.eurosim.info*

✉ *javor@eik.bme.hu*

⌧ HSS / András Jávor,
Budapest Univ. of Technology and Economics,
Sztoczek u. 4, 1111 Budapest, Hungary

HSS Officers	
President	András Jávor, *javor@eik.bme.hu*
Vice president	Gábor Szűcs, *szucs@itm.bme.hu*
Secretary	Ágnes Vigh, *vigh@itm.bme.hu*
Repr. EUROSIM	András Jávor, *javor@eik.bme.hu*
Deputy	Gábor Szűcs, *szucs@itm.bme.hu*
Edit. Board SNE	András Jávor, *javor@eik.bme.hu*
Web EUROSIM	Gábor Szűcs, *szucs@itm.bme.hu*

Last data update March 2008

ISCS – Italian Society for Computer Simulation

The Italian Society for Computer Simulation (ISCS) is a scientific non-profit association of members from industry, university, education and several public and research institutions with common interest in all fields of computer simulation.

→ *www.eurosim.info*

✉ *Mario.savastano@uniina.it*

⌧ ISCS / Mario Savastano,
c/o CNR - IRSIP,
Via Claudio 21, 80125 Napoli, Italy

ISCS Officers	
President	M. Savastano, *mario.savastano@unina.it*
Vice president	F. Maceri, *Franco.Maceri@uniroma2.it*
Repr. EUROSIM	F. Maceri, *Franco.Maceri@uniroma2.it*
Secretary	Paola Provenzano, *paola.provenzano@uniroma2.it*
Edit. Board SNE	M. Savastano, *mario.savastano@unina.it*

Last data update December2010

LIOPHANT Simulation

Liophant Simulation is a non-profit association born in order to be a trait-d'union among simulation developers and users; Liophant is devoted to promote and diffuse the simulation techniques and methodologies; the Association promotes exchange of students, sabbatical years, organization of International Conferences, courses and internships focused on M&S applications.

→ *www.liophant.org*

✉ *info@liophant.org*

⌧ LIOPHANT Simulation, c/o Agostino G. Bruzzone,
DIME, University of Genoa, Savona Campus
via Molinero 1, 17100 Savona (SV), Italy

LIOPHANT Officers	
President	A.G. Bruzzone, *agostino@itim.unige.it*
Director	E. Bocca, *enrico.bocca@liophant.org*
Secretary	A. Devoti, *devoti.a@iveco.com*
Treasurer	Marina Masseimassei@itim.unige.it
Repr. EUROSIM	A.G. Bruzzone, *agostino@itim.unige.it*
Deputy	F. Longo, *f.longo@unical.it*
Edit. Board SNE	F. Longo, *f.longo@unical.it*
Web EUROSIM	F. Longo, *f.longo@unical.it*

Last data update June 2016

LSS – Latvian Simulation Society

The Latvian Simulation Society (LSS) has been founded in 1990 as the first professional simulation organisation in the field of Modelling and simulation in the post-Soviet area. Its members represent the main simulation centres in Latvia, including both academic and industrial sectors.

→ *briedis.itl.rtu.lv/imb/*

✉ *merkur@itl.rtu.lv*

⌧ LSS / Yuri Merkuryev, Dept. of Modelling
and Simulation Riga Technical University
Kalku street 1, Riga, LV-1658, LATVIA

LSS Officers	
President	Yuri Merkuryev, *merkur@itl.rtu.lv*
Secretary	Artis Teilans, *Artis.Teilans@exigenservices.com*
Repr. EUROSIM	Yuri Merkuryev, *merkur@itl.rtu.lv*
Deputy	Artis Teilans, *Artis.Teilans@exigenservices.com*
Edit. Board SNE	Yuri Merkuryev, *merkur@itl.rtu.lv*
Web EUROSIM	Vitaly Bolshakov, *vitalijs.bolsakovs@rtu.lv*

Last data update June 2016

KA-SIM Kosovo Simulation Society

Kosova Association for Modeling and Simulation (KA – SIM, founded in 2009), is part of Kosova Association of Control, Automation and Systems Engineering (KA – CASE). KA–CASE was registered in 2006 as non Profit Organization and since 2009 is National Member of IFAC – International Federation of Automatic Control. KA-SIM joined EUROSIM as Observer Member in 2011. In 2016, KA-SIM became full member.

KA-SIM has about 50 members, and is organizing the international conference series International Conference in Business, Technology and Innovation, in November, in Durrhes, Albania, and IFAC Simulation Workshops in Pristina.

→ *www.ubt-uni.net/ka-case*

✉ *ehajrizi@ubt-uni.net*

⌧ MOD&SIM KA-CASE; Att. Dr. Edmond Hajrizi
Univ. for Business and Technology (UBT)
Lagjja Kalabria p.n., 10000 Prishtina, Kosovo

KA-SIM Officers	
President	Edmond Hajrizi, *ehajrizi@ubt-uni.net*
Vice president	Muzafer Shala, *info@ka-sim.com*
Secretary	Lulzim Beqiri, *info@ka-sim.com*
Treasurer	Selman Berisha, *info@ka-sim.com*
Repr. EUROSIM	Edmond Hajrizi, *ehajrizi@ubt-uni.net*
Deputy	Muzafer Shala, *info@ka-sim.com*
Edit. Board SNE	Edmond Hajrizi, *ehajrizi@ubt-uni.net*
Web EUROSIM	Betim Gashi, *info@ka-sim.com*

Last data update December 2016

PSCS – Polish Society for Computer Simulation

PSCS was founded in 1993 in Warsaw. PSCS is a scientific, non-profit association of members from universities, research institutes and industry in Poland with common interests in variety of methods of computer simulations and its applications. At present PSCS counts 257 members.

→ *www.eurosim.info* (*www.ptsk.man.bialystok.pl)*

✉ *leon@ibib.waw.pl*

⌧ PSCS / Leon Bobrowski, c/o IBIB PAN,
ul. Trojdena 4 (p.416), 02-109 Warszawa, Poland

PSCS Officers	
President	Leon Bobrowski, *leon@ibib.waw.pl*
Vice president	Tadeusz Nowicki, *Tadeusz.Nowicki@wat.edu.pl*
Treasurer	Z. Sosnowski, *zenon@ii.pb.bialystok.pl*
Secretary	Zdzislaw Galkowski, *Zdzislaw.Galkowski@simr.pw.edu.pl*
Repr. EUROSIM	Leon Bobrowski, *leon@ibib.waw.pl*
Deputy	Tadeusz Nowicki, *tadeusz.nowicki@wat.edu.pl*
Edit. Board SNE	Zenon Sosnowski, z.sosnowski@pb.ed.pl
Web EUROSIM	Magdalena Topczewska *m.topczewska@pb.edu.pl*

Last data update December2013

SIMS – Scandinavian Simulation Society

SIMS is the *Scandinavian Simulation Society* with members from the four Nordic countries Denmark, Finland, Norway and Sweden. The SIMS history goes back to 1959. SIMS practical matters are taken care of by the SIMS board consisting of two representatives from each Nordic country (Iceland one board member).

SIMS Structure. SIMS is organised as federation of regional societies. There are FinSim (Finnish Simulation Forum), DKSIM (Dansk Simuleringsforening) and NFA (Norsk Forening for Automatisering).

→ *www.scansims.org*

✉ *esko.juuso@oulu.fi*

⌧ SIMS / Erik Dahlquist, School of Business, Society and Engineering, Department of Energy, Building and Environment, Mälardalen University, P.O.Box 883, 72123 Västerås, Sweden

SIMS Officers	
President	Erik Dahlquist, *erik.dahlquist@mdh.se*
Vice president	Bernd Lie, *lie@hit.noe*
Treasurer	Vadim Engelson, *vadim.engelson@mathcore.com*
Repr. EUROSIM	Erik Dahlquist, *erik.dahlquist@mdh.se*
Edit. Board SNE	Esko Juuso, *esko.juuso@oulu.fi*
Web EUROSIM	Vadim Engelson, *vadim.engelson@mathcore.com*

Last data update June 2016

SLOSIM – Slovenian Society for Simulation and Modelling

SLOSIM - Slovenian Society for Simulation and Modelling was established in 1994 and became the full member of **EUROSIM** in 1996. Currently it has 90 members from both Slovenian universities, institutes, and industry. It promotes modelling and simulation approaches to problem solving in industrial as well as in academic environments by establishing communication and cooperation among corresponding teams.

→ www.slosim.si

🖃 *slosim@fe.uni-lj.si*

🖂 SLOSIM / Vito Logar, Faculty of Electrical
Engineering, University of Ljubljana,
Tržaška 25, 1000 Ljubljana, Slovenia

SLOSIM Officers	
President	Vito Logar, *vito.logar@fe.uni-lj.si*
Vice president	Božidar Šarler, *bozidar.sarler@ung.si*
Secretary	Aleš Belič, *ales.belic@sandoz.com*
Treasurer	Milan Simčič, *milan.simcic@fe.uni-lj.si*
Repr. EUROSIM	B. Zupančič, *borut.zupancic@fe.uni-lj.si*
Deputy	Vito Logar, *vito.logar@fe.uni-lj.si*
Edit. Board SNE	B. Zupančič, *borut.zupancic@fe.uni-lj.si* Vito Logar, *vito.logar@fe.uni-lj.si* Blaž Rodič, *blaz.rodic@fis.unm.si*
Web EUROSIM	Vito Logar, *vito.logar@fe.uni-lj.si*

Last data update December 2016

UKSIM - United Kingdom Simulation Society

The UK Simulation Society is very active in organizing conferences, meetings and workshops. UKSim holds its annual conference in the March-April period. In recent years the conference has always been held at Emmanuel College, Cambridge. The Asia Modelling and Simulation Section (AMSS) of UKSim holds 4-5 conferences per year including the EMS (European Modelling Symposium), an event mainly aimed at young researchers, organized each year by UKSim in different European cities.

Membership of the UK Simulation Society is free to participants of any of our conferences and their co-authors.

→ *www.uksim.org.uk*

🖃 *david.al-dabass@ntu.ac.uk*

🖂 UKSIM / Prof. David Al-Dabass
Computing & Informatics,
Nottingham Trent University
Clifton lane, Nottingham, NG11 8NS
United Kingdom

UKSIM Officers	
President	David Al-Dabass, *david.al-dabass@ntu.ac.uk*
Secretary	A. Orsoni, *A.Orsoni@kingston.ac.uk*
Treasurer	A. Orsoni, *A.Orsoni@kingston.ac.uk*
Membership chair	G. Jenkins, *glenn.l.jenkins@smu.ac.uk*
Local/Venue chair	Richard Cant, *richard.cant@ntu.ac.uk*
Repr. EUROSIM	A. Orsoni, *A.Orsoni@kingston.ac.uk*
Deputy	G. Jenkins, *glenn.l.jenkins@smu.ac.uk*
Edit. Board SNE	A. Orsoni, *A.Orsoni@kingston.ac.uk*

Last data update March 2016

RNSS – Russian Simulation Society

NSS - The Russian National Simulation Society (Национальное Общество Имитационного Моделирования – НОИМ) was officially registered in Russian Federation on February 11, 2011. In February 2012 NSS has been accepted as an observer member of **EUROSIM**, and in 2015 RNSS has become full member.

→ *www.simulation.su*

🖃 *yusupov@iias.spb.su*

🖂 RNSS / R. M. Yusupov,
St. Petersburg Institute of Informatics and Automation
RAS, 199178, St. Petersburg, 14th lin. V.O, 39

RNSS Officers	
President	R. M. Yusupov, *yusupov@iias.spb.su*
Chair Man. Board	A. Plotnikov, *plotnikov@sstc.spb.ru*
Secretary	M. Dolmatov, *dolmatov@simulation.su*
Repr. EUROSIM	R.M. Yusupov, *yusupov@iias.spb.su* Y. Senichenkov, *senyb@dcn.icc.spbstu.ru*
Deputy	B. Sokolov, *sokol@iias.spb.su*
Edit. Board SNE	Y. Senichenkov, *senyb@dcn.icc.spbstu.ru*

Last data update June 2016

EUROSIM OBSERVER MEMBERS

ROMSIM – Romanian Modelling and Simulation Society

ROMSIM has been founded in 1990 as a non-profit society, devoted to theoretical and applied aspects of modelling and simulation of systems. ROMSIM currently has about 100 members from Romania and Moldavia.

→ *www.eurosim.info* (*www.ici.ro/romsim*)

✉ *sflorin@ici.ro*

🖂 ROMSIM / Florin Hartescu,
National Institute for Research in Informatics, Averescu
Av. 8 – 10, 71316 Bucharest, Romania

ROMSIM Officers	
President	
Vice president	Florin Hartescu, *flory@ici.ro* Marius Radulescu, *mradulescu.csmro@yahoo.com*
Repr. EUROSIM	Marius Radulescu, *mradulescu.csmro@yahoo.com*
Deputy	Florin Hartescu, *flory@ici.ro*
Edit. Board SNE	Constanta Zoe Radulescu, *zoe@ici.ro*
Web EUROSIM	Florin Hartescu, *flory@ici.ro*

Last data update partly June 2017

MIMOS – Italian Modelling and Simulation Association

MIMOS (Movimento Italiano Modellazione e Simulazione – Italian Modelling and Simulation Association) is the Italian association grouping companies, professionals, universities, and research institutions working in the field of modelling, simulation, virtual reality and 3D, with the aim of enhancing the culture of 'virtuality' in Italy, in every application area.

MIMOS became **EUROSIM** Observer Member in 2016 and is preparing application for full membership.

→ *www.mimos.it*

✉ *roma@mimos.it – info@mimos.it*

🖂 MIMOS – Movimento Italiano Modellazione e Simulazione; via Ugo Foscolo 4, 10126 Torino – via Laurentina 760, 00143 Roma

MIMOS Officers	
President	Paolo Proietti, *roma@mimos.it*
Secretary	Davide Borra, *segreteria@mimos.it*
Treasurer	Davide Borra, *segreteria@mimos.it*
Repr. EUROSIM	Paolo Proietti, *roma@mimos.it*
Deputy	Agostino Bruzzone, *agostino@itim.unige.it*
Edit. Board SNE	Paolo Proietti, *roma@mimos.it*

Last data update December 2016

CANDIDATES

Albanian Simulation Society

At the Department of Statistics and Applied Informatics, Faculty of Economy, University of Tirana, Prof. Dr. Kozeta Sevrani at present is setting up an Albanian Simulation Society. Kozeta Sevrani, professor of Computer Science and Management Information Systems, and head of the Department of Mathematics, Statistics and Applied Informatic, has attended a EUROSIM board meeting in Vienna and has presented simulation activities in Albania and the new simulation society.

The society – constitution and bylaws are being worked out - will be involved in different international and local simulation projects, and will be engaged in the organisation of the conference series ISTI – Information Systems and Technology. The society intends to become a EUROSIM Observer Member.

✉ *kozeta.sevrani@unitir.edu.al*

🖂 Albanian Simulation Goup, attn. Kozeta Sevrani
University of Tirana, Faculty of Economy
rr. Elbasanit, Tirana 355 Albania

Albanian Simulation Society- Officers (Planned)	
President	Kozeta Sevrani, *kozeta.sevrani@unitir.edu.al*
Secretary	
Treasurer	
Repr. EUROSIM	Kozeta Sevrani, *kozeta.sevrani@unitir.edu.al*
Edit. Board SNE	Albana Gorishti, *albana.gorishti@unitir.edu.al* Majlinda Godolja, *majlinda.godolja@fshn.edu.al*

Last data update December 2016

EUROSIM Federation of European Simulation Societies

General Information. **EUROSIM**, the Federation of European Simulation Societies, was set up in 1989. The purpose of **EUROSIM** is to provide a European forum for simulation societies and groups to promote advancement of modelling and simulation in industry, research, and development.

Member Societies. **EUROSIM** members may be national simulation societies and regional or international societies and groups dealing with modelling and simulation. At present **EUROSIM** has 16 *Full Members* and 2) *Observer Members*, and one member candidate.

→ *www.eurosim.info*

EUROSIM Congresses

The **EUROSIM Congress** can be seen as constant within these developments – each three years simulationists from all over the world gather in one European country to exchange information on development in modelling and simulation.

Report EUROSIM Congress 2016

EUROSIM 2016

9th EUROSIM Congress on Modelling and Simulation

September 12 - 16, 2016, Oulu, Finland

eurosim2016.automaatioseura.fi

The 9th **EUROSIM Congress** on Modelling and Simulation (**EUROSIM** 2016) and the 57th SIMS conference on Simulation and Modelling (SIMS 2016) were held jointly in the City of Oulu, Finland. The *Federation of European Simulation Societies* (**EUROSIM**) was set up in 1989. The purpose of **EUROSIM** is to provide a European forum for regional and national simulation societies to promote the advancement of modelling and simulation in industry, research and development. The congress was organized by *Scandinavian Simulation Society* (SIMS), *Finnish Society of Automation* and *University of Oulu*. SIMS was founded in 1959. The event was co-sponsored by the *IFAC Technical Committees* 3.2, 6.1, 6.2, 6.3 and 6.4, the *IEEE Computer Society* and the *IEEE Finland Section*.

Networking events were organized on three levels: **EUROSIM** had Executive Board Meeting and Board Meetings; **SIMS** had Board Meeting and Annual Meetings and *Finnish Simulation Forum* (**FinSim**), founded in 2001, had Board Meeting and Annual Meetings.

The overall management was done by the **EUROSIM** Board, consisting of representatives of 17 European Simulation Societies lead by the **EUROSIM** President Dr. Esko Juuso. The NOC was chaired by Prof. Kauko Leiviskä with Co-Chair Dr. Esko Juuso and Vice-Chair Industry Dr. Timo Ahola (**FinSim**). The IPC Chair Prof. Erik Dahlquist (**SIMS**), Co-Chair Prof. Bernt Lie (**SIMS**) and Vice-Chair Industry Dr. Lasse Eriksson were leading the IPC which consisted of 49 scientists. The Publication committee was chaired by Dr. Peter Ylén with C-Chairs Prof. Lars Eriksson and Prof. David Al-Dabass. **EUROSIM** and **SIMS** Boards were in the IPC. **EUROSIM** members participated actively: 20 **EUROSIM** countries from 14 member societies participated the congress.

The congress had a multi-conference structure with several special topics related to methodologies, functionalities and application areas. The programme includes invited talks, parallel, special and poster sessions, exhibition and versatile technical and social tours. Plenary presentations covered versatile topics: Thermal management within power engineering, Electric arc furnaces, Autonomous driving and traffic maneuvers, Simulation benefits to the mining, mineral and metal industry, Composition of the atmosphere and biophotonic applications.

Majority of sessions focused on industrial topics, including Applied energy, Solar thermal power plants, Energy supply for buildings, Gasification and power plants, Chemical processes, Capture processes, Manufacturing, Mining, Metal and Oil Industry, Water and wastewater systems, Transportation, Mechatronics and Robotics and Bioprocesses. Methodologies covered Computational intelligence, Conceptual modelling, Complex systems, Data analysis, Discrete event systems, Distributed parameter systems, Parallel and distributed interactive systems and Simulation tools and platforms. In the functionalities, the focus on Control, Optimization, Communication, Security, Education and Training, Human-machine interaction, Planning and scheduling, Sensing, Virtual reality and visualization.

Panel discussions were organised for three topics: Modelling and Simulation in Processes and Cleantech, Future energy systems and Intelligent Systems and IoT in Future Automation.

Four technical tours covered steel industry, energy production, radio technology, printed electronics and mining. Tutorial were arranged on simulation tools and platforms.

The congress was very international: there were 181 participants from 33 countries. There were participants from all the continents, except Africa, but there were co-authors also from Africa. The program included 6 plenaries, 175 regular oral and 7 poster presentations. The focus was in Europe: 1/4 from Finland, 1/5 from other Nordic countries and 1/3 from other European countries.

Conference Site: Oulu City Theatre

Congress Opening – Main Theatre Hall

EUROSIM Board Meeting – 15 delegates from member societies

Taking over EUROSIM President: from Esko Juuso (SIMS, Scandinavia) to Emilio Jiminez (CEA-SMSG, Spain)

Announcement EUROSIM Congress 2016

EUROSIM 2019

10th EUROSIM Congress on Modelling and Simulation

2019, La Rioja, Spain

The **EUROSIM** congress series continues – the 10th EUROSIM Congress, **EUROSIM 2019**, will be organised by **CEA-SMSG** – Spanish Modelling and Simulation Group.

EUROSIM Executive Board announces EUROSIM 2019 (E. Juuso, Past President; E. Jimenez, President and Organiser EUROSIM 2019; F. Breitenecker, Treasurer and SNE Editor-in-Chief; M. Mujica-Mota, Secretary; A. Körner, Board and Web Secretary

E. Juuso, SIMS / EUROSIM,
esko.juuso@oulu.fi

DBSS – Dutch Benelux Simulation Society

General Information. The Dutch Benelux Simulation Society (DBSS) was founded in July 1986 in order to create an organisation of simulation professionals within the Dutch language area. DBSS has actively promoted creation of similar organisations in other language areas. DBSS is a member of **EUROSIM** and works in close co-operation with its members and with affiliated societies.

→ www.eurosim.info

a.w.heemink@its.tudelft.nl

✉ DBSS / A. W. Heemink
Delft University of Technology, ITS - twi,
Mekelweg 4, 2628 CD Delft, The Netherlands

General Activities.

Courses. DBSS will provide the following courses, which are intended to students and professionals:

Discrete event simulation of aviation operations
Lecturer: Miguel Mujica Mota, HvA-Amsterdam
Place and Date; June 20, 2017, 10:00 – 15:00
HvA facilties Aviation cademy, Amsterdam
Requirements: Basic knowledge of Modelling and Simulation and SIMIO

Introduction to R
Lecturer: Ann Wellen, UNAM-Mexico
Place and Date: July 4, 2017, 10-00 – 15:00.
HvA facilities Aviation Academy, Amsterdam
Requirements: Basic knowledge in programming

The courses are certified by the DBSS.
The course fee is 25 Euro for DBSS members and 50 Euro for non members..

Coming Events

Academia and Industry representatives from will be invited to give presentations related to this simulation field in order to start discussions and share ideas with the participants. Academics and industrialists interested in these topics are welcome to attend and contribute.

The symposium will be held in the Snijderszaal at the Faculty of Electrical Engineering, Mathematics and Computer Science at TU Delft, Delft (the Netherlands). September, 8th, as been chosen as provisory date of the symposium. Soon you will receive the preliminar program.

2017 Symposium of the DBSS
TU Delft, The Netherlands
September 8, 2017 (prel.)

For any question about the symposium please contact Paolo Scala (p.m.scala@hva.nl) or Ann Wellens (*wann@unam.mx*). Check the last news about the Symposium at out webpage **www.dutchbss.org**

Announcement of the DBSS members' annual meeting.

Friday, the 8th of September 2017, there will be the annual meeting of DBSS members to discuss about the actual status of the society, and future actions to take. All the DBSS members are invited to take part of it and contribute to the DBSS development. Provisory location is at the Faculty of Electrical Engineering, Mathematics and Computer Science at TU Delft.

.Publications.

Some of our members recently published material that can be of interest for any practitioner:

- Dr. Mujica Mota recently published his book *Applied Simulation and Optimization 2: New Applications in Logistics, Industrial and Aeronautical Practice*, Springer Book. This book deals with the combination of simulation with optimization techniques.
- Dr. Mujica Mota recently published another book *Robust Modelling and Simulation: Integration of SIMIO with Coloured Petri Nets*, Springer Book. This book deals with the integration of SIMIO with CPN for developing models that are easier to reproduce and implement in industry cases.

Various

Internship position
The Aviation Academy together with the Logistics Department of the Amsterdam University of applied Sciences (The Netherlands) is offering a research scholarship in the area of modelling and simulation, the period of the scholarship is one year extensible to another one. The candidate will join the capacity research group at the Amsterdam University of Applied Sciences. It is expected that the candidate participates in the project's research on capacity analysis, in particular *Modelling and Analysis of Air Cargo Logistics Network at Schiphol Airport.*

The ideal candidate should be pursuing or close to starting a MSc in a University, have an excellent background in computer science or industrial engineering or similar, good English language skills (written and spoken), demonstrated background in research activities and real interest in modelling and simulation, interest in aviation operations is a plus. For further information about the research fellowship, please contact Dr. Miguel Mujica Mota at Amsterdam University of Applied Sciences (http://www.hva.nl/aviation /). Complete applications including cover letter, CV, copies of graduate certificates, and a reference letter should be submitted via E-mail in a single file (format pdf) to Dr. Miguel Mujica Mota, Amsterdam University of Applied Sciences, The Netherlands, m.mujica.mota@hva.nl. The application is open until fulfilment of the position.

Post Doc position

The Amsterdam University of Applied Sciences/Hogeschool van Amsterdam (AUAS/HvA, The Netherlands) is looking for an experienced and expert Post Doc in Simulation and Heuristics for capacity in the Aviation Sector.

The position:

As a Post Doc you will contribute to the research on capacity in the aviation sector. Your research project *Air-freight Schiphol: Capacity and future transport of cargo* will aim at evaluating the performance of the future mode of cargo transport in the Airport by important carriers in the Netherlands. Simulation will be the main activity of the project and by the end of the contract algorithmic implementations might be used to implement solutions in the study performed. The objective will be to develop models in order to get insight about the operative restrictions that are imposed by the different elements in the system: weather, transport networks and technical restrictions in general and test novel solutions for transporting the cargo by air. Research will be performed to propose solutions for the identified problems. The project lasts for maximum two years in Amsterdam.

You will conduct research together with colleagues and industry professionals, and present your research results in professional publications, conferences and the university seminar. You will collaborate with other international research groups of recognized prestige.

What we are looking for

We are looking for a doctorate-level expert in simulation with experience preferably in the use of SIMIO and C#, the knowledge in the aviation and/or logistics sector will be highly appreciated. You have a genuine interest in improving the operative problems of the aviation industry through the use of simulation and operations research techniques. You also have excellent verbal and written skills in English. Your research was related to similar topics where simulation and operations research was one of the most important components.

The department/programme

The Aviation Academy was created to serve the European aviation industry. Our mission is to provide the current and next generation of professionals with the skills they need to meet the international aviation challenges of the next 10 to 15 years. Our three main activities are Education, Practical Research and Peer Networking. The Aviation Academy performs practical scientific research around real-world problems in the aviation sector, with the goal of improving and innovating professional practice. As a Post Doc you will participate in all of our three main activities, but your focus will be on practical research.

Within the Aviation Academy the Aviation research programme, in conjunction with the aviation sector, conducts applied research into the optimal development and utilisation of the capacity in the aviation sector. The results of this research will benefit companies in the sector and will be used to enhance the educational programme.

Information

For any questions about this position please contact Dr. Miguel Mujica Mota, assoc. Prof of the Aviation Academy on + 31 06-21158263/ (*m.mujica.mota@hva.nl*).

Please visit *www.DutchBSS.org* for more information.

DBSS, M. Mujica-Mota,
m.mujica.mota@hva.nl

ASIM ASIM ASIM

ASIM - Buchreihen / ASIM Book Series

Proceedings

Simulation in Production und Logistics 2015 - 16. ASIM-Fachtagung Simulation in Produktion und Logistik
M. Raabe, U. Clausen (Hrsg.); ISBN 978-3-8396-0936-1, Stuttgart: Fraunhofer Verlag, 2015. (O. P.)

Simulation in Produktion und Logistik 2013: Entscheidungsunterstützung von der Planung bis zur Steuerung
W. Dangelmaier, C. Laroque, A. Klaas (Hrsg.); ISBN 978-3-942647-35-9, HNI-Verlagsschriftenreihe,
Heinz Nixdorf Institut, Paderborn, 2013 (O. P.)

Modellierung, Regelung und Simulation in Automotive und Prozessautomation – Proc. 5. ASIM-Workshop Wismar 2011. C. Deatcu, P. Dünow, T. Pawletta, S. Pawletta (eds.), ISBN 978-3-901608-36-0,
ASIM/ARGESIM, Wien, 2011 (O. A.)

Simulation in Produktion und Logistik 2010: Integrationsaspekte der Simulation - Technik, Organisation und Personal. G. Zülch, P. Stock, (Hrsg.), ISBN 978-3-86644-558-1, KIT Scientific Publ. Karlsruhe, 2010 (O.P.)

Monographs

Simulation und Optimierung in Produktion und Logistik – Praxisorientierter Leitfaden mit Fallbeispielen.
L. März, W. Krug, O. Rose, G. Weigert , G. (Hrsg.); ISBN 978-3-642-14535-3, Springer, 2011 (O.P.)

Verifikation und Validierung für die Simulation in Produktion und Logistik - Vorgehensmodelle und Techniken.
M. Rabe, S. Spieckermann, S. Wenzel (eds.); ISBN: 978-3-540-35281-5, Springer, Berlin, 2008 (O. P.)

Qualitätskriterien für die Simulation in Produktion und Logistik – Planung und Durchführung von Simulationsstudien. S. Wenzel, M. Weiß, S. Collisi – Böhmer, H. Pitsch, O. Rose (Hrsg.);
ISBN: 978-3-540-35281-5, Springer, Berlin, 2008

Series Fortschrittsberichte Simulation

P. Einzinger: **A Comparative Analysis of System Dynamics and Agent-Based Modelling for Health Care Reimbursement Systems.**
FBS 25, ASIM/ARGESIM Vienna, 2014; ISBN 978-3-901608-75-9, ARGESIM Report 75 (O. A.)

M. Bruckner: **Agentenbasierte Simulation von Personenströmen mit unterschiedlichen Charakteristiken**
FBS 24, ASIM/ARGESIM Vienna, 2014; ISBN 978-3-901608-74-2, ARGESIM Report 74 (O. A.)

S. Emrich: **Deployment of Mathematical Simulation Models for Space Management**
FBS 23, ASIM/ARGESIM Vienna, 2013; ISBN 978-3-901608-73-5, ARGESIM Report 73 (O. A.)

G. Maletzki: **Rapid Control Prototyping komplexer und flexibler Robotersteuerungen auf Basis des SBC-Ansatzes.** FBS 22, ASIM/ARGESIM Vienna, 2014; ISBN 978-3-901608-72-8, ARGESIM Report 72 (O. A.)

X. Descovich: **Lattice Boltzmann Modeling and Simulation of Incompressible Flows in Distensible Tubes for Applications in Hemodynamics**
FBS 21, ASIM/ARGESIM Vienna, 2012; ISBN 978-3-901608-71-1, ARGESIM Report 71 (O. A.)

F. Miksch: **Mathematical Modeling for New Insights into Epidemics by Herd Immunity and Serotype Shift**
FBS 20, ASIM/ARGESIM Vienna, 2012; ISBN 978-3-901608-70-4, ARGESIM Report 70 (O. A.)

S. Tauböck: **Integration of Agent Based Modelling in DEVS for Utilisation Analysis: The MoreSpace Project at TU Vienna;** FBS 19, ASIM/ARGESIM Vienna, 2012; ISBN 978-3-901608-69-8, ARGESIM Report 69 (O. A.)

Ch. Steinbrecher: **Ein Beitrag zur prädiktiven Regelung verbrennungsmotorischer Prozesse**
FBS 18, ASIM/ARGESIM Vienna, 2010; ISBN 978-3-901608-68-1, ARGESIM Report 68 (O. A.)

O. Hagendorf: **Simulation-based Parameter and Structure Optimisation of Discrete Event Systems**
FBS 17, ASIM/ARGESIM Vienna, 2010; ISBN 978-3-901608-67-4, ARGESIM Report 67 (O. A.)

Orders via ASIM (O. A.) **or via Publisher** (O. P.)**:**
ASIM/ARGESIM Office Germany, Hochschule Wismar, PF 1210, 23952 Wismar, Germany
ASIM/ARGESIM Geschäftsstelle Österreich, c/o DWH, Neustiftgasse 57, 1040 Vienna, Austria
Download (some books) via ASIM webpage in preparation

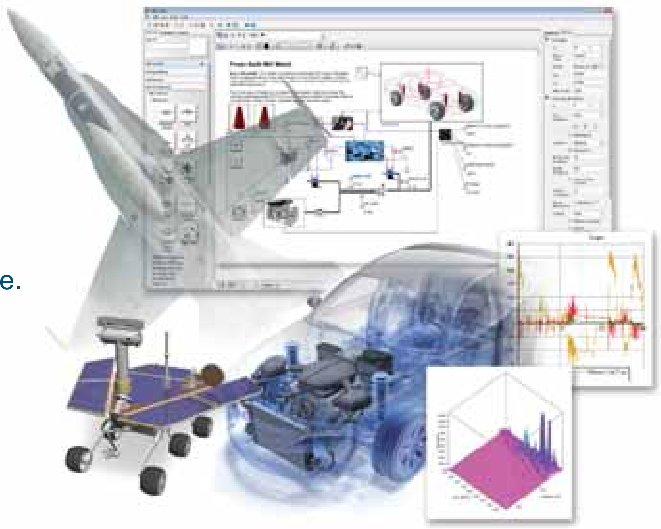
Info: www.asim-gi.org, info@asim-gi.org

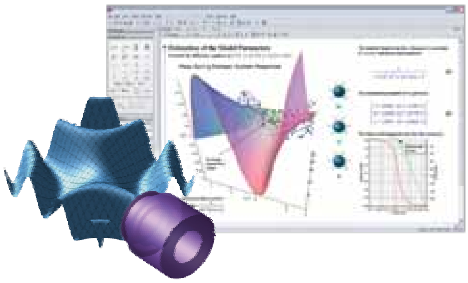
A modern approach to modeling and simulation

MapleSim™

With MapleSim, educators have an industry-proven tool to help bridge the gap between theory and practice.

- MapleSim illustrates concepts, and helps students learn the connection between theory and physical behavior
- A wide variety of models are available to help get started right away





Maple™

MapleSim is built on Maple, which combines the world's most powerful mathematical computation engine with an intuitive, "clickable" user interface.

To learn more about how you can reinforce engineering concepts using a combination of theory, simulation, and hardware, view this webinar.

www.maplesoft.com/SNEWebinar

Contact us: +49 (0)241/980919-30

Maplesoft™
Mathematics • Modeling • Simulation

www.maplesoft.com | vertrieb@maplesoft.com

© 2016 Maplesoft, ein Bereich von Waterloo Maple Inc., 615 Kumpf Drive, Waterloo, ON, N2V1K8, Kanada. Bei Maplesoft, Maple und MapleSim handelt es sich jeweils um Warenzeichen von Waterloo Maple Inc. Alle anderen Warenzeichen sind Eigentum ihrer jeweiligen Inhaber.

www.tuverlag.at
ISBN 978-3-903024-50-2

ISBN 978-3-903024-50-2