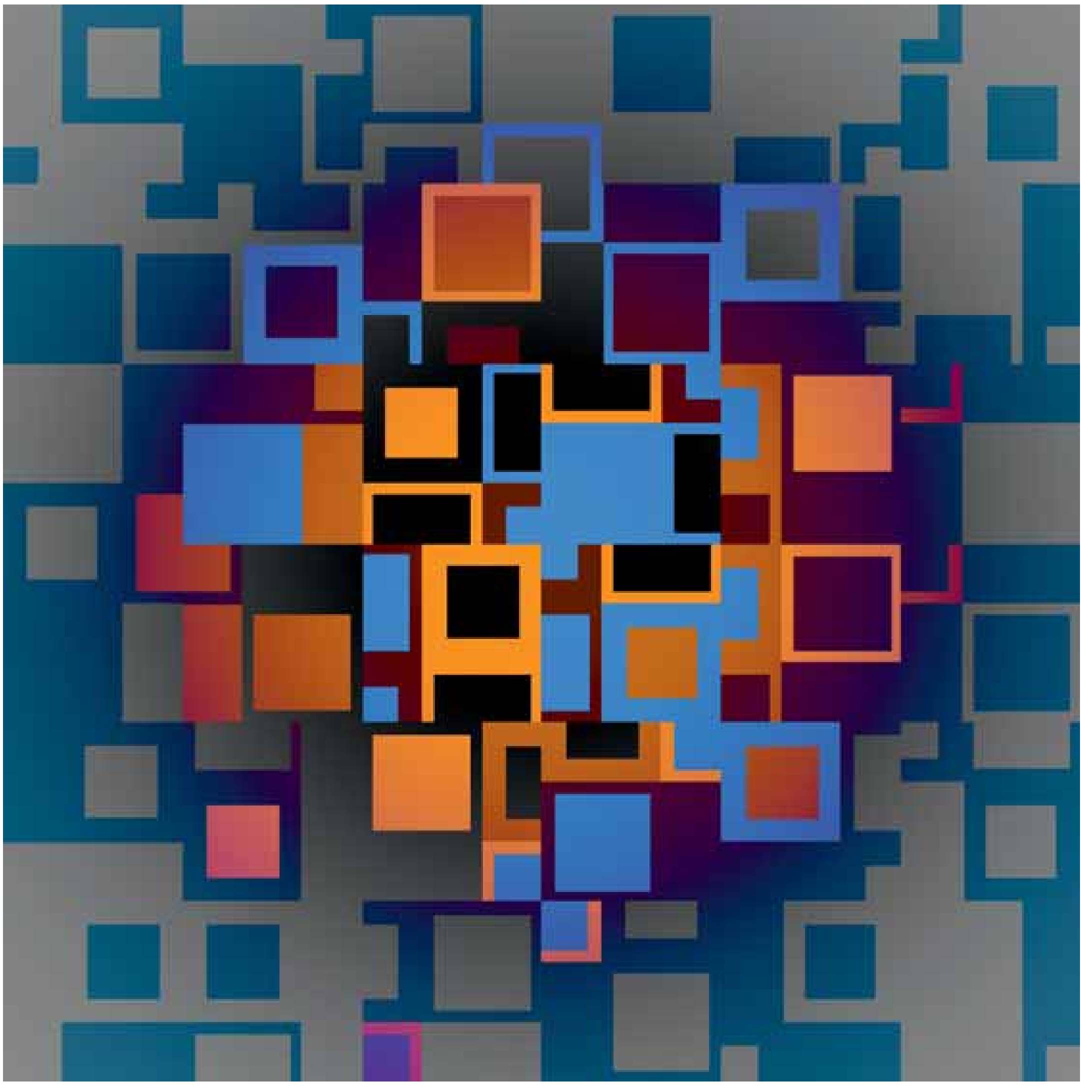
SNE SIMULATION NOTES EUROPE



Journal on Developments and Trends in Modelling and Simulation

EUROSIM Scientific Membership Journal

Vol. 28 No.1, March 2018 ISSN Online 2306-0271 DOI 10.11128/sne.28.1.1040

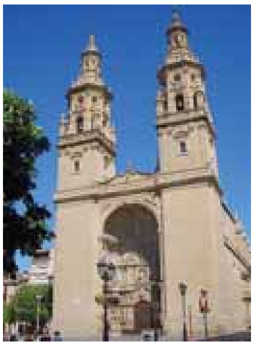
TU Verlag an der Technischen Universität Wien

ARGESIM

EUROSIM 2019

10th EUROSIM Congress on Modelling and Simulation

La Rioja, Logroño, Spain, July 1 – 5, 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

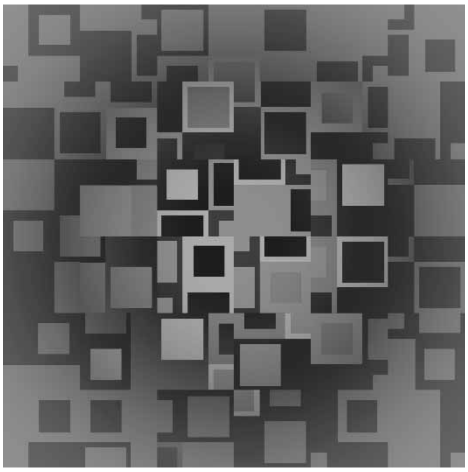
Congress Venue / Social Events The Congress will be held in the City of Logroño, Capital of La Rioja, Northern Spain. The main venue and the exhibition site is the University of La Rioja (UR), located on a modern campus in Logroño, capital of La Rioja, where 7500 students are registered. The UR is the 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.

Congress Team: 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 – We are glad, that for SNE Volume 28 again Vlatko Čerić, past president of CROSSIM, the Croatian Simulation Society, provides his algorithmic art as design for SNE cover pages. 'Algorithms, mathematics and art are interrelated in an art form called algorithmic art. Algorithmic art is visual art generated by algorithms that completely describe creation of images. This kind of art is strongly related with contemporary computer technology, and especially computer programming, as well as with mathematics used in algorithms for image generation' – as Vlatko Čerić defines. The artist and simulationist Vlatko Čerić has chosen four algorithmic art pictures from the series BIRTH for covers of SNE Volume 28 - the technique used is mapping and overlapping of squares which emerge from a centre and which fade out in colour. At right the cover picture for SNE 28(1), the first of the four cover pictures from the series BIRTH. This issue SNE 28(1) again underlines SNE's publication strategy – contributions on developments and trends in modelling and simulation – submitted directly by simulationists for quick publication, and post-conference publication of contributions from EUROSIM conferences, and last but not least contributions dealing with the ARGESIM/EUROSIM benchmarks.

I would like to thank all authors for their contributions to SNE 28(1) showing the broad variety of simulation. 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 co-operation with Vlatko Čerić.

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

Contents SNE 28(1)

Online SNE 28(1), DOI 10.11128/sne.28.1.1040, ARGESIM Publ.
Print SNE 28(1) ISBN 978-3-903024-70-0,
Print-on-Demand, TU Verlag Vienna, www.tuverlag.at

SNE Contact & Info

→ *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

SNE Online ISSN 2306-0271, SNE Print ISSN 2305-9974

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, TU Wien/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*

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** (www.argesim.org).

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.
- **EN** Education Note –6 – 8 p.
- **PN** Project Note 6 – 8 p.
- **SN** Short Note, max. 6 p.
- **SW** Software Note , 4 – 6 p.
- **BN** Benchmark Note, 2 – 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):

A Mesoscopic Bus Transit Simulation Model based on Scarce Data

Daniel Lückerath[1*], Ewald Speckenmeyer[2], Oliver Ullrich[3], Naphtali Rishe[3]

[1]Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS, Schloss Birlinghoven, 53757 Sankt Augustin, Germany; *daniel.lueckerath@iais.fraunhofer.de

[2]Department of Computer Science, University of Cologne, Weyertal 121, 50931 Cologne, Germany

[3]School of Computing and Information Sciences, Florida International University, ECS 243C, 11200 SW 8th St, Miami FL-33199, USA

SNE 28(1), 2018, 1 - 10, DOI: 10.11128/sne.28.tn.10401
Received: July 7, 2017; Revised December 10, 2017;
Accepted December 15, 2017
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna,
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Typically, public transit modeling requires the availability of an extensive data basis to enable detailed modeling, calibration, and validation. Sometimes such data is not available, even though there is demand for a simulation model to examine the impacts of planning decisions and strategies. However, available data may support a simulation model which, while not perfect, at least yields plausible results allowing the examination of broader impacts of planning decisions and strategies.

In this paper a bus transit simulation model custom-tailored to manage on a scarce data basis is described. After an introduction to aims and scope, some background on bus transit systems, data availability, and related research is shared. Then a simulation model utilizing the available scarce data is proposed and the representation of the physical transit network, logical components, vehicle behavior, and transit provider's everyday operational management decisions are described. Finally, the outcomes of initial experiments on a small artificial transit network model are discussed, demonstrating the model's ability to yield plausible results.

Introduction

Typically, public transit modeling requires the availability of a broad data basis, to enable detailed modeling, calibration, and validation.

In some cases, such data is not available, even though there is demand for a simulation model to examine the impacts of planning decisions and strategies. However, available data may support a simulation model which, while not perfect, at least yields plausible results allowing for the examination of broader impacts of planning decisions and strategies.

In this paper an event-based simulation model of bus transit based on scarce data is presented. The model is intended to be used as a tool to evaluate planning decisions and operational management rules to mitigate disturbances in transit networks, e.g. faulty vehicles, blocked stops, or unusually high traffic on street segments. For this intended usage, it is not necessary to include every single behavioral decision in a model, but rather design it in such a way that specific key indicators, e.g. delays, kept connections, or service regularity can be measured. Therefore, the proposed model utilizes a more abstract mesoscopic approach to model part of the vehicle behavior.

Many transit models are developed as an extension of already established models of individual traffic (see e.g. [3], [8], [17]). If these models utilize a fine grained modeling approach, they generally necessitate the availability of an extensive data basis, including detailed information on origin-destination matrices, vehicular dynamics, signaling strategies, and lane changing rules (e.g. see [18]), and include many components which are not immediately interesting for public transit systems. Moreover, employing microscopic models of individual traffic for the examination of large areas (e.g. public transit networks of whole cities) often results in prolonged run times (see e.g. [6], [7]) and the parametrization of individual traffic (see e.g. [17]), reversing the initial modeling decision.

The model proposed in this paper is not based on any pre-existing model of individual traffic, but is newly developed based on the specifics of bus transit, and specifically prepared with the intended use in mind. That, in unison with using a mesoscopic modeling approach, allows for the construction of a simulation model which manages on a very scarce data basis, most of which is publicly available for many transit systems.

The paper continues with some background on bus transit systems, available data, and related research (see section 1). Afterwards a mesoscopic, event-based simulation model of bus transit utilizing scarce data is proposed, describing in detail the representation of the physical transit network, logical components, vehicle behavior, and the transit provider's operational management strategies (see section 2). To evaluate the simulation model's behavior some experiments are conducted based on a model of a small artificial transit network (see section 3). The paper concludes with a summary and an outlook of further research steps (see section 4).

1 Background

1.1 Bus transit networks and available data

A bus transit system consists of a street network and a set of bus stops where passenger exchanges take place. These bus stops are served by a set of transit vehicles executing service trips, i.e. pairings of starting times and sequences of bus stops, according to a timetable. Each individual vehicle executes several service trips, interspersed with deadhead trips, over the course of an operational day, which is called a rotation. Such a rotation usually begins with a deadhead trip from the vehicle's depot to the first stop of its first service trip and, after a number of service trips, ends with a returning deadhead trip to the depot. The vehicle schedule defines the assignment of specific vehicles to rotations.

Some stops are marked as control points, i.e. locations in the network where control strategies may be employed, e.g. purposely delaying early vehicles until the scheduled departure time is reached. At other stops, vehicles depart as soon as the passenger exchange is completed.

Directed paths through the network, connecting two successive stops are called connections. They usually consist of several street segments, junctions, and signals, which in turn can be shared by several connections.

Signals control access to street segments, usually at junctions. Often, two or more signals constitute a signal group with a common scheduling strategy.

Public transit vehicles generally follow pre-defined line routes, consisting of sequences of stops to be serviced.

In most public transit systems, daily operations are managed by an operations center, with dispatcher personnel managing procedures for the mitigation of disturbances originating e.g. from street segments blocked by accidents, or failing transit vehicle doors. These operators have a number of remedies at their disposal, including the authority to short-turn or cancel trips, and to deploy extra vehicles.

The proposed model manages on a minimum of data which is publicly available for many transit systems: a list of stops and their connections, including planned traversal times; timetables for each of these stops; descriptions of the lines and their variants, including the order of stops to be serviced; as well as type and attributes of the vehicles used.

Additional data improves the simulation's accuracy: the distribution of traversal times for each connection and the passenger arrival rate for each stop, both depending on the time of day. In addition, empirical data on individual departure times help with calibration and validation of the model.

1.2 Related research

A number of simulation models covering bus transit can be found in the literature (see e.g. [1], [3], [8], and [15] - [17]).

One of the first models was proposed in 1979 by Andersson et al. in [1]. The authors develop a mesoscopic event-based interactive simulation model for bus transit systems allowing users online testing of operational strategies, like short-turning trips or deadheading vehicles. Andersson et al. model the bus system as a set of lines, i.e. collections of linked stops, where each stop posseses a separate holding bay for every line serving it. As a result, the model does not represent direct vehicle interactions. Instead, interactions between vehicles are modeled indirecly via the passenger exchange process. Because passengers can be served by multiple lines, delays or earliness of a vehicle of one line may affect the passenger exchange processes of vehicles of other lines, resulting in vehicle bunching effects.

The traversal process of vehicles between successive stops is modeled mesoscopically using lognormal distributed random values dependent on the time of day.

A newer mesoscopic approach to bus transit simulation is proposed by Toledo et al. in [17]. The authors extend a mesoscopic simulation model for individual traffic based on queuing theory proposed by Burghout in [4], which represents the street network as a graph of interconnected queues and vehicles as individual entities traversing these queues based on speed/density functions. The nodes of the graph represent junctions, and are modeled as collections of servers, one for each turning movement and each with different processing times based on e.g. the green time ratio of the corresponding signal.

Other recent simulation models including bus transit use microscopic agent-based modeling approaches all of which are based on generic multi-modal transit models (see e.g. [3], [8], [15], and [16]). The open source simulation framework *SUMO* (Simulation of Urban Mobility, for an overview see [2]) started as an agent-based simulation model for individual traffic, representing individual drivers as agents with individual attributes and microscopic vehicle dynamics as well as lane changing behavior. These agents traverse a graph representing the street network, with street segments as edges and junctions as nodes. In [2] and [8] Behrisch et al. and Kendziorra and Weber extend SUMO with the capability to incorporate public transit and individual travellers, respectively. For this purpose, public transit vehicles are modeled as agents of individual traffic with fixed routes, which stop at predefined locations to let passengers board and alight.

A very comprehensive agent-based simulation model including bus transit is proposed by Suzumura et al. in [15] and [16]. They employ the *IBM Mega Traffic Simulator* (see [14]) to develop a parallel, agent-based model for microscopic transit simulation called *M3*. Every single participant and every potential transit mode (e.g. bus, light rail, car, bicycle) is represented by agents, which move through multiple interconnected graphs representing the different transit networks. As in *SUMO*, public transit agents are modeled similar to agents of individual traffic, but with fixed routes on which they have to serve predefined stops.

Apart from the model proposed by Andersson et al., all discussed modeling approaches require large amounts of data about individual traffic and the underlying street network (e.g. origin-destination matrices, signaling strategies). However, when employed to real-world test cases most users of these models choose to parameterize some or all of these model aspects, due to run-time or other concerns (see. e.g. [6], [7], and [17]). Therefore, this paper continues with a new modeling approach taking these considerations into account by basing all modeling decisions on the specifics of public transit systems and simplifying some transit modeling aspects using a mesoscopic approach.

2 Modeling Bus Transit

A public transit system can be decomposed into a number of sub-systems: the physical network consisting of stops, connections and signals, the logical network consisting of lines, rotations, and planned trips, the vehicle sub-system consisting of the buses and their behavior, and the transit provider's operational management decisions. Some of these components show a stochastic behavior and are therefore subject to randomization.

In the following, these system components and their behavior are represented by simulation entities, events and activities, and thus translated to an object-oriented, event-based simulation model (see [19]).

Given the scarcity of the available data, a main goal of the modeling process is to avoid unjustified complexity. While individual vehicles are considered explicitly, their driving dynamics are not modeled in detail, but are subsumed with a certain abstraction.

2.1 Physical network

The physical network is represented by a directed graph $G = (V, E)$, where stops and connections are modeled as nodes $v \in V$, and their neighborhood relations are represented by edges $e \in E$.

A stop $s \in S$ is attributed with an identifier, time of day specific passenger arrival rates and a maximum capacity for concurrently stopping vehicles. In addition, some stops are marked as control points. As stops are assigned to exactly one station, each entity contains a reference to its station object.

Connections $c = (s_i, s_j) \in C \subseteq S \times S$ are directed paths through the transit network between two successive stops s_i and s_j. Connections are attributed with a length, a planned traversal time $t_p(c)$ according to the timetable, and might also contain a list of atomic street segments and a set of signals.

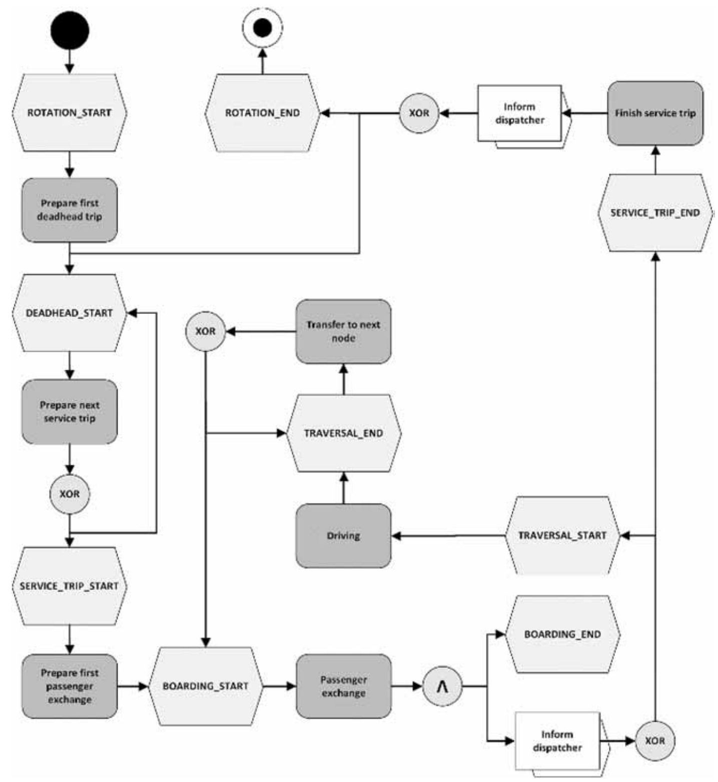


Figure 1: Event-activity-chain for the vehicle sub-model. Hexagons represent events, squares with rounded edges represent activities.

2.2 Logical network

A line $l \in L$ is modeled as an ordered list $l_i = (s_{i_1}, c_{i_1}, s_{i_2}, c_{i_2}, \dots, c_{i_{n-1}}, s_{i_n})$ of stops $s_{i_j} \in S$, which are to be serviced successively by a vehicle of a specific type, interspersed with the relevant connections $c_{i_j} \in C$ between each two successive stops. This avoids elaborate path finding over the course of a simulation run.

Planned service trips are tuples of a line to be served and a planned departure time at the first stop of that line. The set of all service trips defines the services available to prospective passengers during an operational day, i.e. the timetable.

Rotations are ordered sequences of all the trips to be executed by one – not yet specified – vehicle during a single operational day. A vehicle schedule assigns one specific vehicle to each rotation.

2.3 Vehicles

Vehicles are represented by entities traversing the model graph over the course of a simulation run according to timetable, vehicle schedule, and operational strategies, encapsulating a significant part of the simulation logic.

Vehicles are classified according to their type and attributes: the vehicle type defines vehicle length, capacity, maximum velocity, minimum passenger exchange time and exchange rate. These attributes can be overridden by setting values for individual vehicles, enabling both the representation of vehicles of the same type but with different equipment and individual vehicle defects (e.g. lower passenger exchange rates caused by a faulty door).

Each vehicle entity keeps a reference to the node it currently inhabits, i.e. its current position, as well as a reference to the trip it is currently executing. The entity therefore only has knowledge about the environment immediately important to it, all other information, e.g. regarding rotations, timetable or vehicle schedule, is managed by the dispatcher module (see section 2.4).

The vehicle sub-models execute nine types of simulation events (see figure 1): The event types ROTATION_START and ROTATION_END concern the start and end of a vehicle's assigned rotation and associated activities, e.g. notifying the relevant operational management modules. The event type DEADHEAD_START represents the beginning of a deadhead trip.

As the scarce data basis does not include information on actual planned deadhead trips, the model does not execute these trips, but approximates them by following the strategy described in [10] and [12]: after the conclusion of a service trip the vehicle entity is taken out of the model, and right before the planned start of the next planned service trip the entity tries to access the initial stop of the scheduled trip's route. If this node is already filled up to capacity, the respective event is rescheduled until the next vehicle leaves the stop. While this strategy circumvents the missing information, it omits the representation of traffic load generated by deadhead trips. As real-world vehicle schedules are aimed at minimizing costly deadhead trips, the loss of accuracy resulting from this approach is justifiable. The event types SERVICE_TRIP_START and SERVICE_TRIP_END frame the execution of service trips. While the activities triggered by SERVICE_TRIP_START mainly concern preparations for the first passenger exchange, SERVICE_TRIP_END events notify the operational management modules that the vehicle is available to execute further trips. The event types BOARDING_START and BOARDING_END concern the passenger exchange. The passenger exchange time depends on the stop, the vehicle, as well as the time of day and takes the inter-arrival time of successive vehicles into account to model bus bunching (see section 2.5). In addition to being executed as part of the vehicle sub-model, a BOARDING_END event is also sent to relevant operational management modules to allow for the execution of operational strategies. Once the passenger exchange is completed and the operational management module scheduled a departure time, a TRAVERSAL_START event is triggered. The subsequent driving activity is modeled mesoscopically by drawing the necessary traversal time from a random distribution (see section 2.5). An event of type TRAVERSAL_END completes the traversal of the connection. In case the next stop is filled up to capacity, the TRAVERSAL_END event is rescheduled for the predicted time of the blocking vehicle's departure, so the current vehicle can approach the stop.

In case more detailed data is available, the traversal behavior can be represented microscopically, e.g. by dividing the traversal activity into smaller, interconnected activities. An example of this strategy was presented by the authors in [12].

2.4 Operational management

The introduction of operational management modules allows to separate the simulation logic for the execution of a single trip from the simulation logic for the overall organization of an operational day. Furthermore, it allows for easy incorporation of operational strategies used by transportation providers (see e.g. [13]). The simulation model includes three operational management modules: one for vehicle scheduling and fleet management, one for line management, and one for operational decision making and disturbance mitigation. The latter – the dispatcher – constitutes the most important module, encapsulating the simulation logic for the overall organization of the operational day as well as the simulation logic for decisions regarding the operational behavior of vehicles (see figure 2). For this purpose, the module holds data on nearly all model components, including planned and actual timetable as well as vehicle schedule.

The complete operational day is framed by events of the types OPERATIONAL_DAY_START and OPERATIONAL_DAY_END. While executing OPERATIONAL_DAY_START the dispatcher assigns a first trip to execute to each vehicle entity in the vehicle pool, based on the vehicle schedule. Each vehicle's first trip is started by enqueuing an event of the type ROTATION_START. An event OPERATIONAL_DAY_END signals the completion of all service trips, and the return of all vehicles to their respective depots. At each occurrence of an event of type BOARDING_END the module determines statistical indicators for the corresponding vehicle entity, decides on potential operational strategies to carry out, and schedules a TRAVERSAL_START event accordingly. The simplest form of operational strategy only considers the current simulation time t_{sim} and the planned time of departure $t_{dep}(b,s)$ of vehicle b at stop s, and only in case the stop is a control point, thus scheduling the departure at $t = \max\left(t_{dep}(b,s), t_{sim}\right)$. If s is not a control point, the event is scheduled for the current simulation time t_{sim}. The end of a service trip and its signalization to operational management is represented by an event of type SERVICE_TRIP_END. The dispatcher assigns the next trip to be executed to the vehicle entity, and, if necessary, prompts a deadhead trip. If the completed trip is the last trip of the vehicle's rotation, the dispatcher orders it to move to its depot and complete its rotation.

If all trips of all rotations are completed, the dispatcher schedules an event of type OPERATIONAL_DAY_END to effectively end the simulation run.

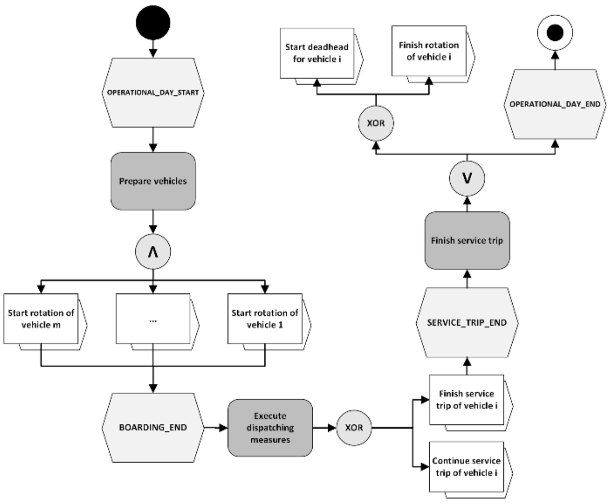


Figure 2: Event-activity-chain for operational management logic.

2.5 Randomization

The described model includes two elements which are influenced by randomization: the vehicle's traversal time for connections, and the passenger exchange times at stops.

Ideally these values are randomized based on distributions whose types and parameters are determined by evaluating empirical data, dependent on location and time of day. As data is scarce in the described scenario though, other methods have to be applied to determine distribution type and parameters.

Traversal times. A lognormal distribution is assumed for the traversal times of a connection $c = (s_i, s_j)$ (see [1]). Lacking detailed data, the parameters of this distribution, i.e. expectancy value and standard deviation, have to be approximated from the planned traversal times $t_p(c)$. These traversal times usually comprise the planned driving time $t_d(c)$ and the planned passenger exchange time $t_b(c)$, which in turn are comprised of average observed driving/passenger exchange times, standard deviations, and unknown terms (see equation 1).

$$\mathrm{t_p}(c) = t_d(c) + t_b(c) = (\mu_c^d + \sigma_c^d + \epsilon_c^d) + (\mu_c^b + \sigma_c^b + \epsilon_c^b) \quad (1)$$

It can be assumed that the planned traversal time $t_p(c)$ is greater than the average observed traversal time μ_c^d to avoid systematic delays. The average traversal time can then be roughly approximated as follows:

$$\hat{\mu}_c^d = t_p(c) * \gamma, \forall c \in C, 0 < \gamma < 1 \quad (2)$$

The ratio γ has to be determined by the user. The standard deviation σ_c^d can be approximated in the same way. It can be assumed that the standard deviation is only a small fraction of the planned traversal time. This yields equation 3.

$$\hat{\sigma}_c^d = t_p(c) * \eta, \forall c \in C, 0 < \eta < 1, \eta \ll \gamma \quad (3)$$

Passenger exchange times. The passenger exchange times are modeled following the method described in [5]. This method is suitable for high frequency transit systems, where it can be assumed that passengers arrive randomly during the inter-arrival time of two successive vehicles, instead of arriving in bulk shortly before the planned departure time. Furthermore, the method facilitates the modeling of bus bunching, i.e. the effect that two vehicles form an undesired platoon because the vehicle in front takes on more passengers than planned and subsequently suffers longer passenger exchange times, while the rear vehicle takes on fewer passengers as planned and thus catches up to the vehicle in front.

If the number $N_{b,s}$ of passengers entering a vehicle b at a stop s, and the average time I_b a passenger takes to enter vehicle b are known, the passenger exchange time $T_{b,s}$ can be determined as follows:

$$T_{b,s} = T_b^{min} + I_b * N_{b,s} \quad (4)$$

Here T_b^{min} describes a vehicle specific minimum time, e.g. for opening and closing the vehicle's doors. If the passenger arrival rate a_s at stop s is known, $N_{b,s}$ can be modeled dependent on the basic interval $T_{L(b)}$ of line $L(b)$ currently served by vehicle b. With $N_{b,s} = T_{L(b)} * a_s$ the passenger exchange time can then be approximated as shown in equation 5.

$$T_{b,s} = T_b^{min} + I_b * T_{L(b)} * a_s \quad (5)$$

If instead of the basic interval between vehicles of the same line, simulated headways between successive vehicles servicing the same stop are used, the model becomes dynamic and thus suitable for a simulation model. If $t_{dep}(b-1, s)$ describes the time a vehicle b's predecessor has serviced the stop, the passenger exchange time $T_{b,s}(t_{sim})$ can be determined as in shown in equation 6.

$$T_{b,s}(t_{sim}) = \begin{cases} T_b^{min}, & b \text{ is first vehicle at } s \\ T_b^{min} + \left(t_{sim} - t_{dep}(b-1, s)\right) * a_s * I_b, & else \end{cases} \quad (6)$$

3 Experiments

Given the scarcity of the available data, an in-depth comparison of the simulation output to real-world data cannot be conducted, and has to be replaced by a theory-driven evaluation (see e.g. [9], p. 206 ff.). Therefore, to evaluate whether the described simulation model yields plausible results, experiments are conducted based on a small artificial bus network. This network, named Universal City Link (UCL, see figure 3), is simple enough for simulation results to be counter-checked by hand, while also constructed to include many of the complications usually found in real-world bus systems (e.g. circular lines and differing basic intervals for different lines). It includes eight lines servicing 40 stops connected by 50 connections.

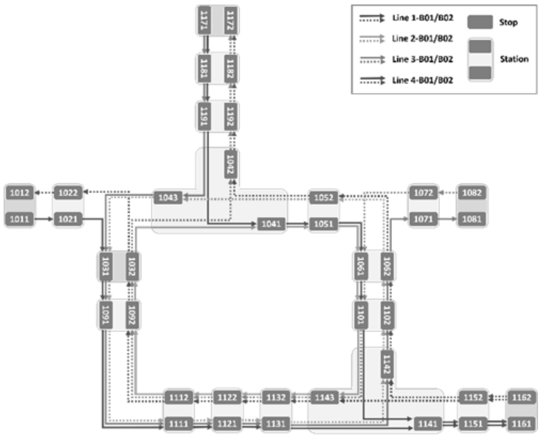


Figure 3: Artificial transit network UCL. Stations colored in darker gray represent starting stations of lines.

Two twelve-hour timetables from 7am to 7pm are compared. Both timetables employ basic intervals of ten minutes for lines 1-B01, 1-B02, 3-B01, 3-B02, 4-B01, and 4-B02 and twenty minute basic intervals for circular lines 2-B01 and 2-B02. For each timetable 100 simulation runs are executed under the same conditions: The first stop of every line is chosen as a control point where vehicles are not allowed to depart before their planned departure time, at all other stops vehicles can depart as soon as the passenger exchange is finished.

The average time a passenger needs to enter a vehicle I_b is set to three seconds for all vehicles, based on random samplings by the authors during their own commutes.

The minimum passenger exchange time T_b^{min} is set to twelve seconds, the time a public transit vehicle usually needs to open and close its doors (see [10]).

The passenger arrival rates a_s are assumed to be constant and chosen in such a way that the average passenger exchange time $T_{b,s}$ equals 20 seconds at every stop, i.e. $T_{b,s} = \bar{T} = 20$. To this end, the values for $\bar{T}$, T_b^{min}, I_b, and $T_{L(b)}$ are inserted into equation 5 and it is solved for a_s. Here, the values for the basic intervals $T_{L(b)}$ are replaced by the theoretically best achivable headways at the different stops of the network, i.e. the equidistanly divided common basic interval at every stop (see equation 7).

$$T_{L(b)} = \left\lfloor \frac{\min\limits_{l_i, l_j \in L(s), l_i \neq l_j} \gcd(T_{l_i}, T_{l_j})}{|L(s)|} \right\rfloor \tag{7}$$

$L(s)$ represents the set of all lines serving stop s and T_{l_i} the basic interval of line $l_i \in L$.

In order to determine values for the ratios γ and η of the traversal times (see equations 2 and 3), the average planned traversal time $\overline{t_p}$ and average passenger exchange time $\bar{T}$ are employed, resulting in $\gamma = 1 - \left(\frac{\bar{T}}{\overline{t_p}}\right) = 1 - \left(\frac{20}{120}\right) \approx 0.83$ and $\eta = 1 - \gamma = 0.17$. Together with the chosen arrival rates, these values should result in rather moderate departure time deviations under both timetables. However, while the first timetable, called UCL+, fits the simulated conditions, i.e. the passenger arrival rates, well, the second timetable, called UCL-, does not fit the simulated conditions. Accordingy, the simulation results should allow to identify a better performance under timetable UCL+ than under timetable UCL-, despite only moderate departure deviations.

The examination of the average departure deviation under both timetables confirms these assumptions: Under timetable UCL- delayed departures on average deviate 11.9s from their planned departure time, while they on average only deviate 8.5s under timetable UCL+, a reduction of 28.6%. Simultaneously, early departures under timetable UCL- on average deviate 3.9s from their planned departure times, while the deviation of early departures under timetable UCL+ is on average 4.7s, an increase of 20.5%. Accordingly, timetable UCL+ exhibits more early departures, while timetable UCL- shows a higher number of delayed departures (see figure 4): Under timetable UCL- 2,699 of 4,740 departures (56.9%) are late, of which 2,175 (45.9%) exhibit a delay of 30s or less. The number of early departures under timetable UCL- is 1,391 (29.4%), of which 1,258 (26.5%) do not exceed a deviation of 30s.

In contrast, under timetable UCL+ 2,361 of 4,740 departures (49.8%) are late, with 2,034 (42.9%) exhibiting a delay of 30s or less. 1,700 departures (35.9%) under timetable UCL+ are early, of which 1,575 (33.2%) are at most 30s early.

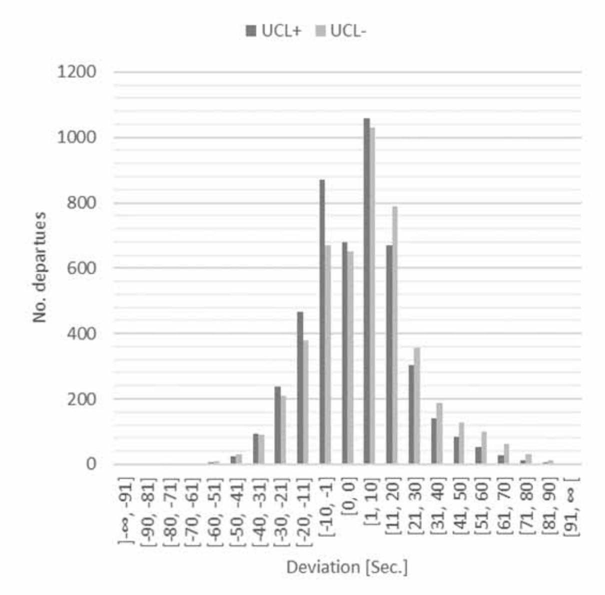


Figure 4: Departure deviation frequency distribution.

Analysing the observed vehicle bunching effects under both timetables, their difference in suitability becomes more distinct. Vehicle bunching effects are measured during simulation runs via the cumulative relative headway redaction ratio, which measures the cumulative percentage of observed headways exhibiting a certain amount of reduction relative to their scheduled value.

As can be seen in figure 5 both timetables exhibit roughly the same amount of overall headway reduction, with timetable UCL- having a 2.01% higher reduction than timetable UCL+. However, under timetable UCL+ the bulk of the reduced headways (16,5%) stay under ten percent, while under timetable UCL- the mayority of reduced headways exhibit a reduction of ten percent or more. And while there is virtually no headway reduction of 50% or more under timetable UCL+, 7.60% of reduced headways under timetable UCL- exhibit such a significant reduction. This suggests that timetable UCL- more heavily suffers reliability issues than timetable UCL+, verifying that the latter is better suited to the simulated conditions and that the simulation model can indeed be used to evaluate the suitability of a timetable for certain simulation conditions.

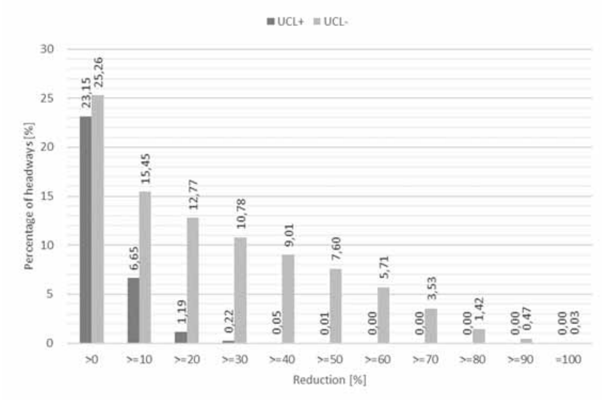


Figure 5: Cumulative relative headway reduction ratio.

To further ensure the plausibility of the simulation model, an examplary analysis of the departure deviation development of line 1-B01 under both timetables is conducted. To this end, the average, median, maximum, and minimum departure deviation as well as its 25%- and 75%-quantile is measured at every stop along the route of line 1-B01. As can be seen in figures 6 and 7 the departure deviation development of line 1-B01 shows roughly the same pattern under both timetables, reaching the highest delay at stop 1121, three stops after line 1-B01 joins lines 2-B02 and 3-B01 at stop 1031. Under timetable UCL+ the delay at stop 1121 is 14.2s on average, while it is 9.6s under timetable UCL-. This is due to a higher (planned) headway between vehicles of line 1-B01 and their predecessors between stops 1031 and 1131 under timetable UCL+ than under timetable UCL- (four minutes versus three minutes), subsequently resulting in a slightly higher number of passengers boarding vehicles of line 1-B01 under timetable UCL+, provoking vehicle bunching effects and prolonging the passenger exchange time.

Lastly, figures 6 and 7 reveal another interesting phenomenon: The range between the 25%- and 75%-quantile is significantly larger under timetable UCL- than under timetable UCL+. This is due to the difference in basic intervals between lines 1-B01 and 2-B02 and the fact that vehicles of line 1-B01 are scheduled to depart three minutes after vehicles of line 2-B02 at stop 1031 under timetable UCL-, while they are scheduled to depart four minutes after vehicles of line 3-B01 under timetable UCL+. Consequently, every second vehicle of line 1-B01 is subject to systematically higher headways under timetable UCL-, namely every time no vehicle of line 2-B02 departs. On the other hand, this phenomenon is not present for vehicles of line 1-B01 under timetable UCL+, but instead for vehicles of line 3-B01.

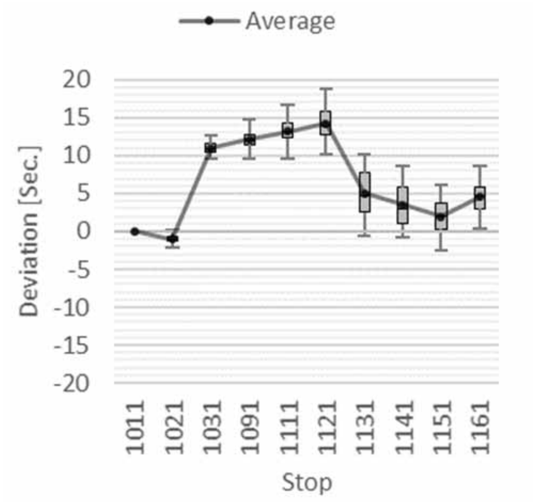


Figure 6: Departure deviation development, line 1-B01, timetable UCL+.

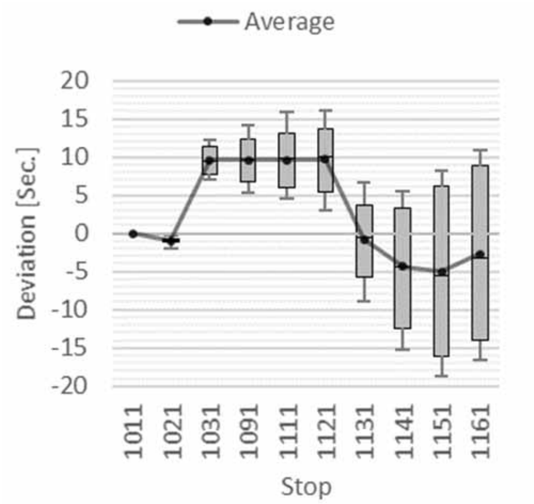


Figure 7: Departure deviation development, line 1-B01, timetable UCL-.

In summary, the results show that the simulation model is suitable to represent bus transit systems. The examined indicators suggest that the model shows plausible behavior under both timetables, even under scarce data.

4 Conclusions

This paper presented a mesoscopic event-based simulation model of bus transit systems. The model consists of sub-models for the representation of the physical network, the logical network, transit vehicles, and the transit provider's operational management. The model is designed to manage on scarce data, especially publicly available time table and network data.

Even though a data-driven evaluation was not feasible without a more comprehensive data basis, the theory-based evaluation based on a model of a small artificial transit network demonstrated the model's plausible behavior.

In further steps, the simulation model first will be applied to models of real world transit systems, before being extended to include light rail transit and further rule-based methos to mitigate disturbances in multimodal transit networks.

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] Andersson P, Hermansson A, Tengvald E, Scalia-Tomba G. Analysis and simulation of an urban bus route. *Transport. Res. A-Pol.*, 1979, 13(6): 439-466.

[2] Behrisch M, Bieker L, Krajzewicz D. SUMO – Simulation of Urban Mobility – an Overview. In: Omerovic, A., Simoni, D., Bobashev, G., editors. SIMUL 2011: The Third International Conference on Advances in System Simulation. *Third International Conference on Advances in System Simulation*; 2011 Oct; Barcelona. Barcelona: IARIA. 55-60.

[3] Behrisch M, Erdmann J, Krajzewicz D. Adding intermodality to the microscopic simulation package SUMO. In: Al-Akaidi, M., editor. MESM' 2010 – GAMEON-ARABIA'2010. 11th Middle Eastern Simulation Multiconference; 2010 Dec; Alexandira. Alexandira: eurosis. 59-66.

[4] Burghout W. *Hybrid microscopic-mesoscopic traffic simulation* [dissertation]. Department of Infrastructure, Royal Institute of Technology, Sweden. University of Stockholm, 2004.

[5] Chapman R, Michel J. Modelling the Tendency of Buses to Form Pairs. *Transport. Sci.*, 1978, 12(2): 165-175

[6] Kastner K, Keber R, Pau P, Samal M. Real-Time Traffic Conditions with SUMO for ITS Austria. In: Behrisch M, Knocke M, editors. 1st SUMO User Conference 2013. *1st SUMO User Conference*; 2013 May, Berlin. Berlin: Deutsches Zentrum für Luft- und Raumfahrt, Institut für Verkehrssystemtechnik. 43-53.

[7] Kastner K, Pau P. Experiences with SUMO in a Real-Life Trafic Monitoring System. In: Behrisch M, Weber M, editors. SUMO 2015 – Intermodal Simulation for Intermodal Transport. *3rd SUMO User Conference*; 2015 May. Berlin: Deutsches zentrum für Luft- und Raumfahrt, Institut für Verkehrssystemtechnik. 1-10.

[8] Kendziorra A, Weber M. Extensions for logistics and public transport in SUMO. In: Behrisch M, Weber M, editors. SUMO 2015 – Intermodal Simulation for Intermodal Transport. *3rd SUMO User Conference*; 2015 May. Berlin: Deutsches zentrum für Luft- und Raumfahrt, Institut für Verkehrssystemtechnik. 83-90.

[9] Liebl F. *Simulation*. 2nd edition. Munich: Oldenbourg Pub.; 1995. 273p.

[10] Lückemeyer G. *A Traffic Simulation System Increasing the Efficiency of Schedule Design for Public Transport Systems Based on Scarce Data* [dissertation]. Department of Computer Science. University of Cologne, 2007.

[11] Lückerath D. *Ein Simulationsmodell für Öffentlichen Personennahverkehr mit regelbasiertem Verkehrsmanagement* [dissertation]. Department of Computer Science. University of Cologne, 2017.

[12] Lückerath D, Ullrich O, Speckenmeyer E. Modeling time table based tram traffic. *SNE*. 2012, 22(2): 61-68. doi: 10.11128/sne.22.tn.10121.

[13] Lückerath D, Ullrich O, Speckenmeyer E. Applicability of rescheduling strategies in tram networks. In: Reichardt R, editor. Proceedings of ASIM-Workshop STS/GMMS 2013. *ASIM-Workshop STS/GMMS*; 2013 Feb; Düsseldorf. Vienna, Austria: ARGESIM/ASIM Pub. 57-64.

[14] Osogami T, Imamichi T, Mizuta H, Morimura T, Raymond R, Suzumura T, Takahashi R, Idé T. Research Report IBM Mega Traffic Simulator. Technical Report, 2012.

[15] Suzumura T, Kanezashi H. Multi-modal traffic simulation platform on parallel and distributed systems. In: Tolk A, Diallo S, Ryzhow I, Yilmaz L, Buckley S, Miller J. Proceedings of the 2014 Winter Simulation Conference. *Winter Simulation Conference*; 2014 Dec; Savannah. Piscataway, NJ, USA: IEEE Press. 769-780.

[16] Suzumura T, McArdle G, Kanezashi H. A high performance multi-modal traffic simulation platform and its case study with the Dublin city. In: Yilmaz, L., Chan, W., Moon, I., Roeder, T., Macal, C., Rossetti, M., editors. Proceedings of the 2015 Winter Simulation Conference. *Winter Simulation Conference*; 2015 Dec; Huntington Beach. Piscataway, NJ, USA: IEEE Press. 767-778.

[17] Toledo T, Cats O, Burghout W, Koutsopoulos H. Mesocopic simulation for transit operations. *Transport. Res. C-Emer*. 2010; 18(6): 896-908.

[18] Ullrich O, Proff I, Lückerath D, Kuckertz P, Speckenmeyer E. Agent-based modeling and simulation of individual traffic as an environment for bus schedule simulation. In: Busch, F., Spangler, M., editors. ITS for Connected Mobility. *mobil.TUM*; 2014; Munich. Munich: Schriftenreihe des Lehrstuhls für Verkehrstechnik der Technischen Universität München. 89-98.

[19] Ullrich O, Lückerath D. An Introduction to Discrete-Event Modeling and Simulation. *SNE*. 2017, 27(1).

A Framework for the Metamodeling of Multi-variant Systems and Reactive Simulation Model Generation and Execution

Thorsten Pawletta[1*], Artur Schmidt[1], Umut Durak[2], Bernard P. Zeigler[3]

[1]Wismar University of Applied Sciences, Research Group Computational Engineering and Automation (CEA), Philipp-Müller-Str. 14, D-23966 Wismar, Germany, **thorsten.pawletta@hs-wismar.de*

[2]German Aerospace Center (DLR), Institute of Fligth Systems, Lilienthalplatz 7, 38108 Braunschweig, Germany

[3]RTSync Corp. 4425 E Agave Rd. Suite 106 Phoenix, AZ 85044, USA

SNE 28(1), 2018, 11 - 18, DOI: 10.11128/sne.28.tn.10402
Received: May 9, 2017 (Selected ASIM STS 2016 Postconf. Publ.), Accepted: December 15, 2017
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. The simulation-based study of Cyber-Physical Systems or complex production systems leads often to a vast number of system variants. Each system variant is characterized by a particular model structure and parameter settings, although system variants may also share common parts. There are two main approaches for modeling such a set of system variants. On the one hand, all variants are mapped in a big model with variation points and on the other hand variants are specified on a higher level of abstraction using a metamodel that is processed with appropriate transformation methods. This paper proposes an approach for modeling system variants using the System Entity Structure (SES) Ontology. It introduces new concepts and advances the SES by a procedural knowledge specification. Moreover, it proposes a software infrastructure for the automated and reactive generation and execution of simulation models based on a SES in combination with a model base. Finally, it refers to a prototype implementation within MATLAB/Simulink and forward-looking within Python.

Introduction

The study of multifaceted end user requirements of Cyber-Physical Systems or of multi-variant production systems leads to a vast number of system variants. Both problem types can be considered as a variability problem.

Variability has been defined as the ability of a system or an artefact to be configured, customized or extended for employment in a particular context [1]. In software engineering Software Product Lines (SPL) are widely employed for developing systems that are characterized by a high degree of variability. SPL define variation points where different variants of products can be derived for varying requirements [2]. Variability management has also been introduced as a challenge to be tackled for model-based testing architectures [3], for model-based concept development tools [4] and for studying multi-variant production systems [5] or reactive robot controls [6]. In this context the problem of reactiveness is also discussed [5, 6, 7]. We will consider reactiveness as the generation and execution of a new system variant depending on current and previous results.

Variability mechanisms shall be defined at particular levels of abstraction, ranging from metamodeling to implementation of the source code. Using a metamodel for variability modeling requires appropriate model transformation methods for mapping to the execution level. This is a particular challenge, because such methods are not supported by the established modeling and simulation environments used in the engineering or production system domain. Another approach is the combination with software tools for requirement or variant management [8] or with domain oriented tools [4]. However, in this case often different kinds of models have to be maintained and kept consistent.

For these are reasons, in engineering and production system applications variability is still often encoded within the executable system models. However, these models are often hard to manage.

Therefore, specific modularization and configuration methods have been developed to tackle the complexity. From the simulation theory the approach of dynamic structure or variable structure systems [9, 10] is known. On the other side, rather pragmatic solutions have been developed, such as for the MATLAB/Simulink environment in [11, 12].

For the modeling and simulation of modular, hierarchical systems, Zeigler introduced the *System Entity Structure* (SES) for specifying a set of system configurations, called a family of systems. The SES approach has evolved steadily to an ontology for model and data engineering [13, 14]. In combination with a model base (MB), organizing a set of configurable basic models, the SES approach has been advanced to a modeling and simulation framework (SES/MB) [15]. In this paper, a reworked version of [16], we extend the SES ontology by adding new features. In addition, we advance the SES/MB framework to an infrastructure for reactive model generation and simulation execution and we refer to a prototype implementation. Using an exemplified multi-variant engineering problem, a concrete SES model, which is a metamodel, is developed. Based on the example, fundamental elements and axioms of the baseline SES ontology are briefly summarized. Next, some extensions to the SES ontology are discussed. The main new features are SESVariables and SESFunctions, which expand the SES ontology by procedural knowledge elements. After that, the selection of a concrete system variant from an SES metamodel is considered and the whole procedure for generating an executable simulation model is depicted under the aspect of reactiveness.

1 Multi-variant Engineering Example

The example is an extension of an application that has been introduced by The MathWorks in [12] to demonstrate features for variant modeling within MATLAB/Simulink. We use that example to make our approach comparable with The MathWorks solution for experienced users. The substantial problem statement is illustrated in Figure 1a. Different controller (ctrl) designs, based on a linear (lc) or a nonlinear (nc) control structure, should be investigated using different signal sources from a signal generator (sg).

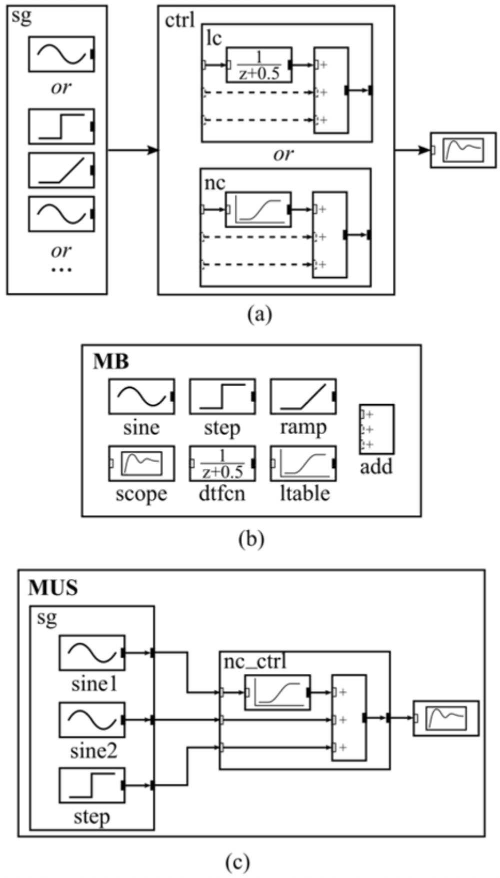


Figure 1: **(a)** Overall engineering problem with a set of system variants; **(b)** Blocks from the Simulink blockset representing the MB; **(c)** Model structure of a concrete system variant.

In addition to the control structure, the signal types {sine | ramp | step} and the number of signal sources {1…3} may vary. Figure 1a shows the two control approaches (lc_ctrl | nc_ctrl) as alternative submodels of model ctrl. Due to the varying number of possible input signals, both approaches lead to three different internal model structures. The minimal internal structure of a ctrl model with one input signal is illustrated with full lines. The extension for two or three input signals is pictured with dashed lines. In the same manner, the internal structure of the system generator (sg) depends on the number and type of included signal sources. Overall, the exemplary problem comprises $(3^1+3^2+3^3)*2$ various system structures. All possible system structures can be aggregated using 7 basic systems. In this case, the basic systems are blocks from the Simulink blockset, which represent a *model base* (MB), as shown in Figure 1b.

Figure 1c illustrates as an example the model structure of a specific system variant, which we call a model under study (MUS). In this case, the MUS consists of an nc_ctrl model, which is influenced by a sg model with three signal sources. Two sources are of type sine and one of type step. For simplification purposes, a separation between MUS and experimental frame (EF) according to [15] is not considered.

2 Metamodel-Based Variant Modeling

This section describes the specification of the exemplary problem to demonstrate multi-variant modeling using an SES. The specification is based on the baseline SES definitions in [13], but it uses some modifications based on former works in [17] and introduces some new concepts, such as the *SESFunctions*.

2.1 SES fundamentals and SES variables

The SES ontology supports the description of a family of systems regarding their elements and the relations between them. It is axiomatically defined and can be represented as a directed labeled tree. Nodes are divided into two types, *entity* and *descriptive* nodes, which can define specific *attributes*. Entity nodes describe system elements and the system itself (root node). The leaf nodes are always entity nodes, whose attributes define a link to a basic model in the MB (attribute *mb*) and possible parameter settings for the referenced basic model. Descriptive nodes express relationships between entities and are subdivided into: *aspect, specialization* and *multi-aspect* nodes.

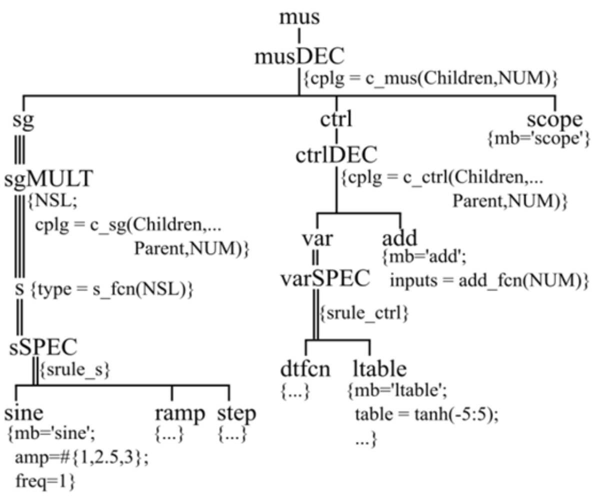


Figure 2: SES tree for the example in Figure 1a

The SES axioms will be considered subsequently, as necessary for the example. Figure 2 illustrates an SES tree that maps the problem described in Section 2. In the tree descriptive nodes are marked with name suffixes: (i) DEC for aspect, (ii) SPEC for specialization and (iii) MULT for multi-aspect. At this point the SES axiom *alternating mode* for entity and descriptive nodes should be noted.

Before describing the SES tree in detail, the new concept of *SESVariables* as the input interface of an SES is explained. This new feature was introduced to support the integration of an SES metamodel, referring to the metamodel definition in [18], in the later suggested infrastructure. In the infrastructure the selection of a particular system variant depends on the current settings of SESVariables. The selection procedure itself is described in the next section. SESVariables have a global scope and are written in uppercase letters in the tree. Two SESVariables in the tree in Figure 2 are defined as input arguments and a third one as an auxiliary variable. They have the following definitions:

```
SESVariables = {SPEC_CTRL,NSL} with
SPEC_CTRL ∊ {'lc','nc'}
NSL ∊ {(i),(i,j),(i,j,k) |
 i ∊ {'sine[x]','ramp','step'}∧
 j ∊ {'sine[x]','ramp','step'}∧
 k ∊ {'sine[x]','ramp','step'}∧
 x ∊ {1,2,3}}
auxiliarySESVariable = {NUM } with
NUM = numel(NSL)
```

According to the exemplary problem (see Fig. 1), the variable *SPEC_CTRL* encodes the desired control structure and the variable *NSL* specifies a list with the signal sources to be selected. The index value *x* allows the encoding of different parameter selections for a *sine* signal. The auxiliary variable *NUM* calculates the current number of elements (`numel`) in *NSL*. An example for an admissible value assignment to SESVariables is given as follows.

```
NSL = {'sine[1]','sine[2]','step'}
SPEC_CTRL = {'nc'}
→ NUM = 3
```

2.2 Decomposition of systems with variable coupling relations

The system itself *(mus)* is represented in the SES tree with the root node. The subsequent aspect *musDEC* and vertical lines define a decomposition of *mus* (parent) in the entities *sg*, *ctrl* and *scope* (children).

The aspect attribute *{cplg=…}* defines the coupling relations of *mus*. Model couplings can be divided into internal couplings (IC) between children, and external input as well as external output couplings (EIC, EOC) between the parent and its children. However, a coupling relationship always has the following structure:

```
{'SrcEntity','FromPrt','SinkEntity',
 'ToPrt'}
```

In the example some ICs of entity *mus* depend on the number of signal sources defined by *sg* (see Fig. 1). To express such dynamics with minimal effort and to keep a lean SES tree, the concept of *SESFunctions* has been introduced. The SESFunctions are like ordinary functions. They extend the declarative specification defined by the baseline SES by procedural knowledge descriptions. SESFunctions are calculated during the processing of an SES, called *pruning*, and are described in the next section. This means that the terms:

```
cplg = c_mus (Children,NUM)
cplg = c_ctrl(Children,Parent,NUM)
```

represent ordinary function calls that return the coupling relations, which depend on the current settings of the input arguments. The variables *Children* and *Parent* are implicit attributes of each tree node, which save the names of the successor (left-to-right) and predecessor nodes. Hence, the set of variable couplings of entity *mus*, derived from the overall problem illustrated in Figure 1, can be defined using the following SESFunction (in MATLAB syntax):

```
function cplg = c_mus(children,num)
%create empty data structure for
%couplings
cplg = cell(num+1,4);
%set variable ICs btwn sg & ctrl
for i=1:num %for 1 to num
 cplg(i,1:4)={children{1},num2str(i),…
             children{2},num2str(i)};
end
%set fixed IC btwn ctrl & scope
cplg(num+1,1:4)={children{2},'1',…
                children{3},'1' };
end
```

The children *sg* and *ctrl* of *mus* are composed entities, while *scope* is an atomic entity. Leaf node *scope* maps a basic system in the SES and defines with its attribute *mb='scope'* a corresponding link to the MB. The decomposition of entity *ctrl* in the entities *var* and *add* is specified by its successor node *ctrlDEC*. In both control approaches, the linear and nonlinear (see Fig. 1), the coupling relations of *ctrl* depend on the number of external inputs, which again depend on the current number of signal sources. Thus, the coupling relations at *ctrlDEC* are specified by an SESFunction analogous to node *musDEC*.

2.3 Variable system attributes and the specialization of systems

Leaf node *add* represents a basic model, such as node *scope*. In contrast to *scope*, it defines a variable attribute for parameter settings, using the SESFunction call `inputs=add_fcn(NUM)`. As illustrated in Figure 1, the configuration of *add* depends on the number of inputs. This problem is specified with the following SESFunction (MATLAB syntax):

```
function inputs = add_fcn(num)
   inputs(1)='|'; inputs(2:num+1)='+';
end
```

The characteristic of entity node *var* is specified by the succeeding specialization node *varSPEC*, marked with double-line edges. A specialization describes an *is-a-relation* concerning the succeeding nodes; in this case, entity *var* can be *dtfcn* or *ltable*. While processing an SES, the selection is controlled by evaluating selection rules that are specified as node attribute. In this case the following rule is defined.

```
srule_ctrl = {
     SPEC_CTRL=='lc' → dtfcn |
     SPEC_CTRL=='nc' → ltable }
```

For specializations the specific SES axiom *inheritance* is defined. Its effects will be explained in the next subsection. The leaf nodes *dtfcn* and *ltable* represent once again basic models. The node *ltable* shows a further example for a variable attribute definition.

2.4 Variable decomposition of systems

According to the problem description in Section 2, the node *sg*, following the aspect *musDEC*, represents a system entity composed of a variable number of signal sources of various types. Referring to the baseline SES definition, such selection and composition has to be specified using a combination of aspect or multi-aspect and specialization nodes, possibly supplemented by selection constraints. However, this approach quickly leads to a confusing SES tree. In the following, an approach for keeping a lean SES tree will be described.

In former work [17] regarding concepts of SES, a first idea for solving this specific problem was discussed under the constraint of relaxing the SES *strict hierarchy* axiom.

Based on this idea, we will suggest a complete solution without the violation of the strict hierarchy axiom. In Figure 2 the entity *sg* is characterized by the succeeding multi-aspect *sgMULT* with triple-line edges. According to the baseline SES definition, a multi-aspect is a special case of an aspect in which the succeeding entities are homogeneous in nature. Thus, it has only one succeeding entity node and defines the number of replications of this node as an attribute. Accordingly, the node *sgMULT* has one succeeding entity node named *s*. However, the node attribute definitions of the multi-aspect *sgMULT* and the succeeding entity *s* are more complex referring to the baseline SES definition. Node *sgMULT* specifies in the SESVariable *NSL* a list of types for replication. The number of replications is implicitly specified by the number of list elements. Remember the example

```
NSL = {'sine[1]','sine[2]','step'}
```

stated at the end of Subsection 3.1. Furthermore, *sgMULT* defines variable coupling relations using the SESFunction call `cplg=c_sg(…)`, analogous to the aspect nodes *musDEC* and *ctrlDEC*.

The entity *s* specifies an attribute *type*. The concrete value of this attribute is determined by calling the SESFunction `s_fcn(NSL)` when processing the SES. The SESFunction `s_fcn` defines a simple iterator.

```
function [type] = s_fcn(NSL)
   persistent idx %static variable
   %init iterator
   if isempty(idx), idx=1; end
   type=NSL(idx); idx=idx+1;
   %reinit iterator
   if numel(NSL)==idx, idx=1; end
end
```

Thus, for each replication of entity *s* an individual value assignment is made, such as in our supposed case *type='sine[1]'*, *type='sine[2]'* and *type='step'*, when processing the SES. Based on the setting of attribute *type*, replications of entity *s* can be specialized using a succeeding specialization node. This is specified in the SES tree with the node *sSPEC*, which defines the various signal sources as succeeding entities and the following selection rule as its attribute.

```
srule_s = {
 Parent.type=='sine[x]'→ sine[x] |
 Parent.type=='ramp'    → ramp |
 Parent.type=='step'    → step     }
```

This means that the selection at *sSPEC* depends on the value assignment to attribute *type* at the parent node of *sSPEC*. Details of this subject will be discussed in the next section (see Fig. 4).

The leaf node entities *sine, ramp* and *step* once again represent basic systems, which specify a link to the MB and parameter configurations. The attribute *amp=#{1,2.5,3}* of entity *sine* defines an ordered multiset for different parameter configurations. Therefore, specifications referring to a *sine* signal source are extended by the index *x* to choose an element from the multiset *amp*.

3 Selecting a Distinct System Variant

An SES, such as in Figure 2, codes a set of system variants and is a metamodel referring to the definition in [18]. For simulation studies a single or several distinct system variants must be derived from the SES metamodel. The selection of a particular model structure, including parameter settings for basic models, depends on the current settings of SESVariables and the selection itself is performed by graph *pruning*. The result of pruning is a decision-free tree, called *Pruned Entity Structure* (PES), which contains all of the necessary knowledge for building a distinct simulation model using basic models from the MB. Figure 3 shows one PES derived from the SES in Figure 2 using the subsequent value assignments to the SESVariables.

```
NSL = {'sine[1]','sine[2]','step'}
SPEC_CTRL = {'nc'}
→ NUM = 3
```

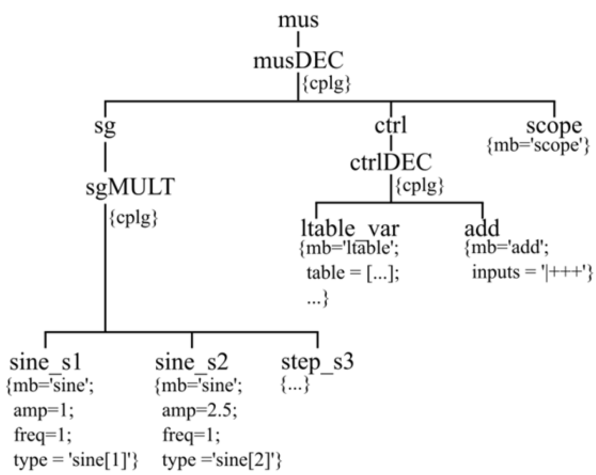


Figure 3: PES derived from SES in Figure 2

The PES in Figure 3 codes a system structure analogous to the MUS in Figure 1c. Subsequently, we will describe the pruning operation in detail. Starting at the root node of the SES in Figure 2, the first decision operation occurs at aspect *musDEC*. The SESFunction called *cplg=c_mus(Children,3)* is executed to determine the coupling relations. The result is:

```
musDEC.cplg = {
  'sg'   , '1' , 'ctrl' , '1';
  'sg'   , '2' , 'ctrl' , '2';
  'sg'   , '3' , 'ctrl' , '3';
  'ctrl', '1' , 'scope', '1' }
```

The next decision point is at multi-aspect *sgMULT*. According to the number of elements in SESVariable *NSL*, the entity *s*, including its following sub-tree, has to be replicated three times. During this operation replicas of *s* are renamed to comply with the *valid brothers* axiom. Moreover, any replica is assigned an exact value to its attribute *type* by executing the iterator SESFunction *s_fcn* with the input argument *NSL={'sine[1]','sine[2]','step'}*. The results of this operation are the replicated and renamed entities *s1*, *s2, s3* with their identical sub-trees but an individual value assignment to their attribute *type*, as illustrated in Figure 4. Now, for each entity s_i the replicated sub-tree is evaluated. This means that the selection rule *srule_s* is evaluated for each node *sSPEC*. In our case, it delivers the following selection *sine[1], sine[2]* and *step*. Remember, the indices of *sine* denote the parameter selection for the multiset of attribute *amp*.

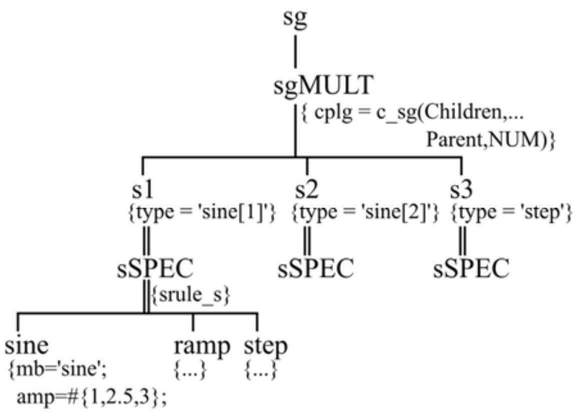


Figure 4: Part of 'intermediate PES' (sub-tree of sg) during pruning of multi-aspect sgMULT

Next, the parent and child entity of each specialization relation is combined according to the *inheritance* axiom. In this case, only the entity names and attributes have to be combined, e.g. *sine_s1{mb='sine';…; type='sine[1]'}*. Finally, the current coupling relations, specified at node *sgMULT*, are determined by executing the SESFunction call *cplg=c_sg(Children, Parent, NUM)*. The result is:

```
sgMULT.cplg = {
  'sine_s1', '1', 'sg', '1';
  'sine_s2', '1', 'sg', '2';
  'step'   , '1', 'sg', '3' }
```

The sub-tree of entity *ctrl* in Figure 2 is resolved in a similar manner during pruning. The resulting coupling relations for entity *ctrlDEC* are the following:

```
ctrlDEC.cplg = {
  'ctrl'      , '1', 'ltable_var','1';
  'ltable_var', '1', 'add'       ,'1';
  'add'       , '1', 'ctrl'      ,'1';
  'ctrl'      , '2', 'add'       ,'2';
  'ctrl'      , '3', 'add'       ,'3' }
```

As mentioned in the beginning, the PES contains all of the necessary knowledge for building a simulation model using basic models from the MB. Sometimes, the PES contains unnecessary attributes due to the pruning operation, such as *type* in the entities *sine_s1* and *sine_s2*, which can be neglected when building the simulation model. Moreover, referring to [15], the PES can be *flattened* by restructuring. Then, in our case the inner nodes *sg, sgMULT, ctrl, ctrlDEC* are resolved and all coupling relations are restructured in the *cplg* attribute of aspect *musDEC*.

4 Software Infrastructure and Prototype Implementation

Figure 5 shows the proposed infrastructure for multi-variant modeling and reactive model generation and execution. The core element is the SES/MB framework according to [15], which is extended by an input and output interface using the introduced SESVariables. This part of the infrastructure maps the functionality as described in the previous sections: (i) basic models are organized in an MB; (ii) the set of system variants is specified in an SES; (iii) the selection of a particular system variant depends on the current settings of SESVariables, it is performed by the pruning operation and its result is a decision-free tree structure, called PES.

Then, an executable simulation model (EM) can be generated based on the PES and basic models from the MB using an appropriate *translation* method. The composition of EM as tuple (MUSi, EFj) means that it consists of a Model Under Study (MUS) and a corresponding Experimental Frame (EF), according to the theory in [15, 19]. The indices i and j are markers for a certain system configuration. An EM is transmitted to the *Execution Unit* (EU). The EU performs three major tasks: (i) linking an EM with a simulation engine; (ii) executing a simulation run; and (iii) collecting the results.

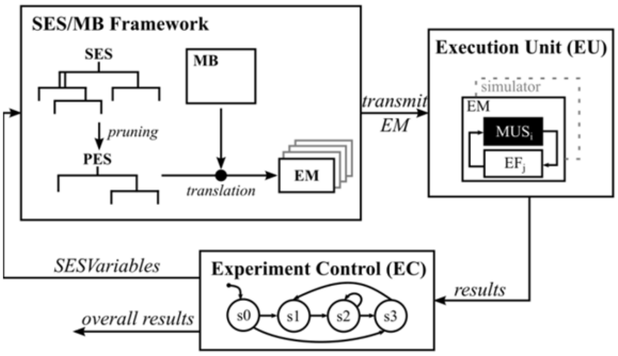


Figure 5: Infrastructure for multi-variant modeling and reactive model generation and execution.

Once the execution phase is complete, the results are sent to the *Experiment Control* (EC). The EC manages the order of EM generations and executions. Based on an application-dependent algorithm, the EC computes new settings for SESVariables as current input for the SES/MB framework to start the next cycle. Additionally, the EC collects all intermediate results from the EU. By means of the feedback loop, structural changes of a variable structure system or experiments with several system configurations can be executed in a reactive manner. Finally, the EC provides the overall results to the user or another software component.

For the investigation of multiple system configurations, such as in our introduced engineering example, it can be useful to generate an EM suite, as illustrated in Figure 5, and to execute it in a sequential or distributed manner by the EU.

The proposed infrastructure has been implemented in the MATLAB/Simulink environment. Thus, a parallel or distributed execution of EMs by the EU is directly supported. Basic implementation aspects of the infrastructure within MATLAB/Simulink and its usage for solving a specific class of multi-variant problems are discussed in [20].

A core element of the infrastructure is the SES toolbox for MATLAB/Simulink, which has been developed by the Research Group CEA [21, 22]. The toolbox provides a graphical SES editor and several methods, such as: (i) *merging* to synthesize various SESs'; (ii) *pruning* for deriving a PES; (iii) *flattening* for the hierarchy reduction of a PES; (iv) *validity checking* of an SES and PES; and (v) *translation* scripts or templates to build EM for Simulink or MATLAB/DEVS [23]. Advanced engineering applications for deploying the SES toolbox for MATLAB/Simulink in the field of model-based testing can be found in [20]. Moreover, a new prototype of the SES toolbox, implemented with Python and supporting an XML interface, is in development to open the way for investigating the approach in conjunction with other simulation environments.

5 Conclusion

Multi-variant modeling and reactive model generation and execution is an important requirement in systems and production engineering. This paper presented a metamodel-based approach using the SES ontology and introduced an appropriate infrastructure to solve this requirement. In addition to the baseline SES definition, the approach uses some new extensions which have been explained step by step based on an engineering example. The introduced concept of SESFunctions advances the declarative knowledge representation through a procedural knowledge specification. Particularly for the modeling of systems with a high degree of variability, the SESFunctions support maintaining a lean SES even for complex problems.

In a next step, this assumption has to be proven by applying the approach to more complex examples. The discussed infrastructure, implemented within MATLAB/Simulink, provides a basis for solving more complex engineering problems. Currently, it is used for developing the reactive and structure variable controls of interacting industrial robots and in the field of objective fidelity evaluation of flight and research simulators. Moreover, a new prototype of the SES toolbox, implemented with Python and supporting an XML interface, is in development to open the way for investigating the approach in conjunction with other simulation environments.

ACKNOWLEDGMENTS

The authors, T. Pawletta and A. Schmidt, gratefully acknowledge the grant from the German Science Foundation DFG (PA 631/2).

References

[1] Capilla R, Bosch J, Kang KC. *Systems and Software Variability Management.* Springer Pub., 2013.

[2] Polzer A, et al. *Managing complexity and variability of a model-based embedded software product line.* Innovations System Software Engineering. 2012, Vol. 8, 35-49.

[3] Sargardui G, Etxeberria L, Agirre JA. *Variability management in testing architectures for embedded control systems.* In: The 4th Int. Conf. on Advances in System Testing and Validation Lifecycle (VALID 2012). IARIA Press, 2012, 73-78.

[4] Krausz M, Zimmer M, Reuss HC. *OverNight Testing – The fully automated simulation environment for evaluation of car concepts.* SNE – Simulation Notes Europe, 2014, Vol. 24(2), 87-94.

[5] Schmidt A, Pawletta T.: *Ein Ontologie-basierter Modellierungs- und Simulationsansatz am Beispiel der ressourceneffizienten Planung spanender Prozessketten.* In: Proc. 15. ASIM Fachtagung Simulation in Produktion und Logistik, Paderborn, HNI-Verlagsschriftenreihe Bd. 316, 2013, 481-490.

[6] Schwatinski T, Pawletta T, Pawletta S. *Flexible Task Oriented Robot Controls Using the System Entity Structure and Model Base Approach.* SNE – Simulation Notes Europe, 2012, Vol. 22(2), 107-114.

[7] Zander J. *Model-based Testing of Real-Time Embedded Systems in the Automotive Domain.* PhD Thesis, TU Berlin, 2008.

[8] Beuche D. *Modeling and building software product lines with pure::variants.* In: Proc. 12th Int. Software Product Line Conf., IEEE Comp. Soc., 2008, 358-358.

[9] Barros FJ. *The dynamic structure discrete event system specification formalism.* Transactions of the SCS International, 1996, No.1, 35-46.

[10] Pawletta T, Lampe B, Pawletta S, Drewelow W. *A DEVS-based approach for modeling and simulation of hybrid variable structure systems.* In: Modeling, Anlysis, and Design of Hybrid Systems, Lecture Notes in Control and Information Sciences 279, Springer Pub. , 2002, 107-129.

[11] Haber A. et al. *First-class variability modeling in MATLAB/Simulink.* In: Proc. of the 7th Int. Workshop on Variability Modelling of Software-intensive Systems. ACM Press, 2013, 11-18.

[12] The MathWorks. *Variant management.* http://de.mathworks.com/help/simulink/ug/variant-management.html, Date of reading 12-Nov. 2015.

[13] Zeigler BP, Hammonds PE. *Modeling and Simulation-Based Data Engineering.* Academic Press, 2007.

[14] Zeigler BP, Sarjoughian HS. *Guide to Modeling and Simulation of Systems of Systems.* Springer Pub., 2012.

[15] Zeigler BP, Prähofer H, Kim TG. *Theory of Modeling and Simulation.* Academic Press, 2000.

[16] Pawletta T, Schmidt A, Zeigler BP, Durak U. *Extended Variability Modeling Using System Entity Structure Ontology within MATLAB/ Simulink.* In: Proc. SprigSim-ANSS, 2016, Pasadena, USA, 62-69.

[17] Rozenblit JW, Zeigler BP. *Representing and constructing system specifications using the system entity structure concepts.* In: Proc. of the Winter Simulation Conf., 1993, 604-611.

[18] Pidcock W, Uschold M. *What are the differences between a vocabulary, a taxonomy, a thesaurus, an ontology, and a meta-model?* In: The Web Graph Database, http://infogrid.org/trac/wiki/ Reference/PidcockArticle, Date of reading 7-July 2016.

[19] Traoré M, Muzy A. *Capturing the dual relationship between simulation models and their context.* Simulation Modelling Practice and Theory, 2006, No. 2, 126-142.

[20] Schmidt A, Durak U, Pawletta T. *Model-based testing methodology using system entity structures for MATLAB/Simulink models.* Simulation: Transactions of SCS Int., Sage Pub., 2016, Vol. 92(8), 729-746.

[21] Pawletta T, Pascheka D, Schmidt A, Pawletta S. *Ontology-assisted system modeling and simulation within MATLAB/Simulink.* SNE – Simulation Notes Europe, 2014, Vol. 24(2), 59-68.

[22] Pawletta T, Schmidt A, Pascheka D. *SES Toolbox for MATLAB/Simulink.* https://www.mb. hs-wismar.de/cea/SES_Tbx/, Date of reading 12-Nov. 2015.

[23] Deatcu C, Schwatinski T, Pawletta T. *DEVS Toolbox for MATLAB.* https://www.mb.hs-wismar.de/cea/SES_Tbx/, Date of reading 12-Nov. 2015.

Let's Build a Tunnel! – A Closer Look at Cologne's New Subway Routes

Oliver Ullrich[1*], Daniel Lückerath[1], Naphtali Rishe[2], Ewald Speckenmeyer[3]

[1]Fraunhofer Institute for Intelligent Analysis and Information Systems, Schloss Birlinghoven, 53757 Sankt Augustin, Germany, *oliver.ullrich@iais.fraunhofer.de

[2]School of Computing and Information Sciences, Florida International University, ECS 243C, 11200 SW 8th St, Miami FL-33199

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

SNE 28(1), 2018, 19-26, DOI: 10.11128/sne.28.tn.10403
Received: January 10, 2018; Revised: March 8,2018
Accepted: March 15, 2018
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Once planning and construction will be complete, Cologne's new subway tunnel will enable fast and direct transportation between the central and southern urban quarters, and the city's central station. This paper takes a closer look at the project's four decades spanning history, its characteristics, and the impact of its integration on the overall system's performance.

When representations of the tunnel and the re-routed light-rail lines are integrated with an existing simulation model, experiments show that no negative impact on the network's performance is to be expected. Some weaknesses are discovered, resulting for example in delayed departures at the tunnel's stations.

Introduction

In the year 1992 the City of Cologne, Germany, decided to build a new subway route under its city center. Construction of the 6.6 kilometer route was planned to be executed in three phases (see Figure 1): during the first phase a new subway tunnel would be built under the city center, creating a fast and direct connection of the central and southern urban quarters to Cologne's central station; during a second phase this tunnel would be extended to connect to existing tracks north of station Schönhauser Strasse at the river Rhine; and as a third phase the tracks leaving the southern end of the tunnel would be continued further south with five new stations and a Park and Ride station being added. When construction began in 2004 the complete system was planned to be operational in 2011.

Almost from the beginning the construction process was ridden with incidents: In September 2004 residents living next to the construction site woke up one morning to find that the tower of the neighborhood church St. Johann Baptist was visibly tilted (see [2]). In November 2004 significant construction-related damage was detected in arches and ceiling of the St. Maria im Kapitol church (see [11]). In August 2007 the tower of the Historic City Hall was found to have shifted (see [21]). Also in summer 2007 a natural gas pipeline was damaged in the course of the constructions, resulting in evacuations and enforced electricity cut-offs in major parts of the city center (see [18]).

Finally, on March 3, 2009 a foundational wall in the 25 meter deep excavation at Waidmarkt square broke, causing the Cologne Historical Archive building and two neighboring residential buildings to collapse into the construction pit, killing two residents and burying 30 shelf kilometers of historical records documenting 1,200 years of local and regional history (see [2]).

When it became known that only 20 percent of the mandated steel joists had been used in the construction of the foundational wall, and that instead of the permitted three a total of 15 well pumps had been installed to keep a much higher than expected volume of water from flowing back into the pit, the state secretary of transportation called the affair "obviously criminal," requesting swift and thorough investigations (see [4]). Eight years later, at the time of this writing, still no-one was indicted in connection with the incident (see [10]).

In the aftermath of the disaster 519 buildings along the construction site were checked, with approximately 300 of them showing significant damage caused by the tunnel's construction (see [19]). Later, when the first vehicles traversing the tunnel caused vibrations in Cologne Cathedral's foundations, a low maximum speed was prescribed for the tunnel segments adjacent to the historic landmark. This issue was allegedly solved by the installation of rubber dampers (see [3]).

At the time of this writing parts of the new subway routes have commenced operation, with the central part around the still open Waidmarkt excavations and the southern extensions still missing. The City now plans the completion of construction and the start of full operations for 2023 (see [13]).

This paper takes a closer look at Cologne's new subway tunnel – its characteristics, its integration with the light-rail network, and potential bottlenecks. To examine the tunnel's expected impact on the network's overall performance a simulation model for timetable-based tram traffic first proposed in [15] is applied.

The paper continues with an overview of the applied simulation model, the tunnel's characteristics and its integration with the model (see Section 1), and then discusses a set of experiments designed to examine the impact the tunnel's integration might have on system performance (see Section 2). The paper concludes with a short summary of the lessons learned (see Section 3).

1 Modeling Cologne's central subway tunnel

1.1 Simulation model

Cologne's light-railway system is mixed – trams travel on underground tracks as well as on street level, and are thus subject to individual traffic and corresponding traffic regulation strategies. Most rail-bound traffic simulations are designed for long distance train or railway networks (see e.g. [17], [20]). While those systems feature similarities to tram networks (see [7], [9], and [22]), e.g. passenger exchange or maneuvering capabilities, they differ greatly in other aspects, e.g. the continuous use of safety blocks.

Subsequently, the applied model (described in detail in [15]) represents tram behavior as a mixture between train and car behavior, e.g. line-of-sight operating and driving. The mixed tram network is modeled as a directed graph with platforms, tracks and track switches represented by nodes. Neighborhood relations between these elements are represented as edges. Figure 3 shows part of the examined network, which is mapped on the graph depicted in Figure 4, where rectangles represent platforms, lines represent tracks and triangles track switches. Stations are defined as sets of geographically related platforms that are connected by walkable infrastructure.

Figure 1: Planned route of Cologne's new subway tunnel, stations, interconnections, and line routes.

The operational logic of transit vehicles is encapsulated in agents (see [16]), with the simulation engine's mechanics being based upon the event-oriented approach (see [1] or [25]). Thus agents change their state while executing simulation events of certain types at discrete points in simulation time. These state changes may trigger a change in the overall system state and generate follow-up events that are fed to appropriate agents. Main tram characteristics are specified by the type of tram, which holds functions for the maneuvering capabilities, e.g. acceleration and braking. The simulation's main stochastic parameters are the probability p_d of introducing small delays in any acceleration activity, and the triangular distribution parameters $a_{h,v}$, $b_{h,v}$, and $c_{h,v}$ for the duration of passenger exchange (see Figure 2), which are specific to platform h and tram type v. Here, the combined duration of opening and closing the vehicle doors m_v serves as a minimum value (see [14]).

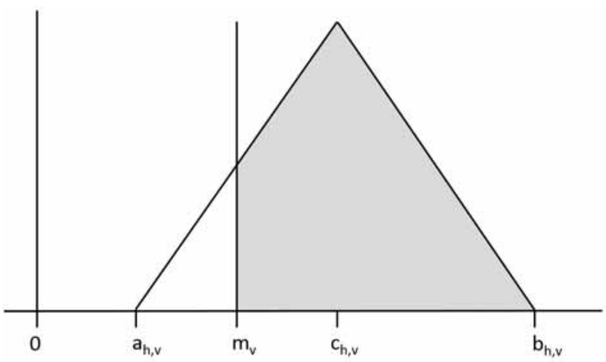


Figure 2: Density of passenger loading time distribution with minimum m_v and triangular distribution determined by parameters $a_{h,v}$, $b_{h,v}$, and $c_{h,v}$.

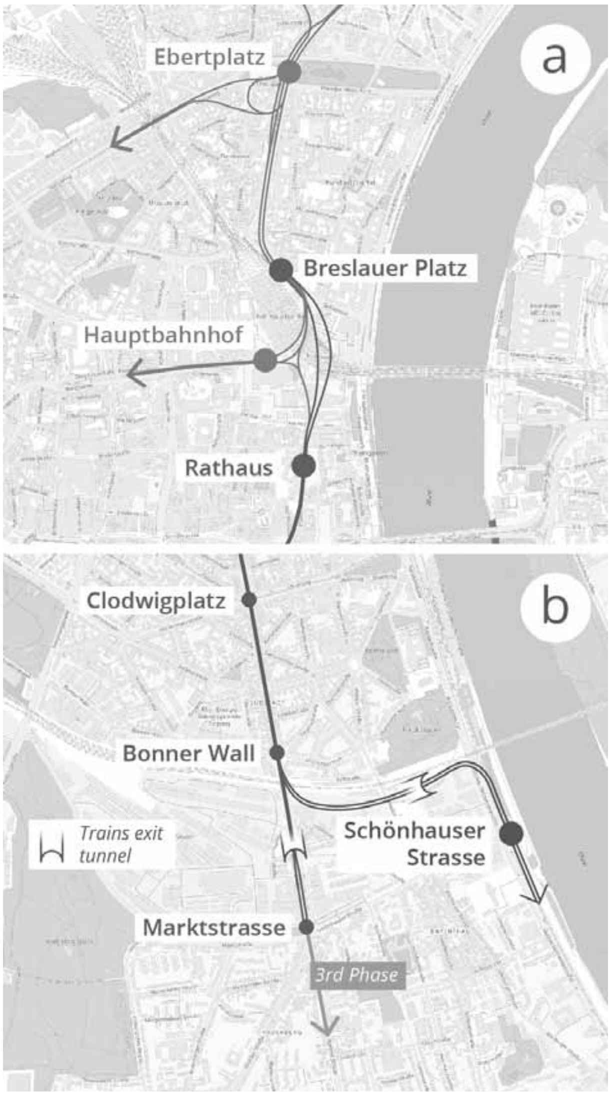


Figure 3: Planned integration with the existing light-rail network in the north (a) and south (b).

1.2 Cologne's central subway tunnel

The tunnel itself (for the construction-related information presented in this sub-section see [12]) is approximately 4 kilometers long and positioned between 11.5 and 28.5 meters underground. It is lined by eight new or significantly extended stations (see Figure 1): Breslauer Platz, Rathaus, Heumarkt, Severinsstrasse, Karthäuserhof, Chlodwigplatz, Bonner Wall, and Marktstrasse. The low average distance between two stations of approximately 570 meters is justified by the high housing density in Cologne's city center.

At its northern end the tunnel is connected to the existing network at the extended station Breslauer Platz, positioned between stations Ebertplatz, Hauptbahnhof, and Rathaus (see Figure 3 (a)). A cluster of switches lies between Ebertplatz und Breslauer Platz – these have to be navigated in configurations dependent on a vehicle's route. Another set of switches lies between Breslauer Platz, Hauptbahnhof, and Rathaus. As switches typically only allow low maximum traversal speed, and are additionally shared between vehicles of different lines, they potentially turn out to be bottlenecks in transit systems (see [26]).

At its southern end the tunnel splits between stations Bonner Wall and Marktstrasse, with one branch continuing south servicing a yet to be built chain of stations towards station Arnoldshöhe, and the other branch leading east where it integrates to the existing network via another set of switches north of the existing station Schönhauser Strasse at the river Rhine's bank (see Figure 3 (b)). Here, another potential bottleneck arises with the new tracks crossing the major artery Gustav-Heinemann-Ufer at ground level, necessitating either wating times for the transit vehicles, or a transit signal priority system with frequent waiting periods for individual traffic on the highway – an option that might be politically hard to justify. The alternative option of an extended tunnel to the east side of Gustav-Heinemann-Ufer, which would have eliminated the ground level crossing, was not realized – primarily for budget reasons.

To service the added stations some line routes are changed: Line 5 leaves its current route to enter the tunnel at station Hauptbahnhof, traverses it, services stop Bonner Strasse, and then goes on to station Arnoldshöhe in the south. Line 16 leaves its old route north of station Schönhauser Strasse, and services the stations Bonner Wall, Chlodwigplatz, Kartäuserwall, Severinsstrasse, Heumarkt, and Rathaus before exiting the tunnel at Breslauer Platz. This rerouting relieves the existing east-west tunnel under the town center around Neumarkt station, which line 16 traverses under the current schedule. Line 16 is complemented by a trunk line 16A servicing the stops between Ebertplatz in the north and Marktstrasse at the tunnel's south end, enabling a higher service frequency in these densely populated areas. These three routes are planned to operate in ten minute intervals each, resulting in tunnel stops being serviced every three to four minutes.

To represent these features each platform, split, and track segment is integrated as a node in the existing model graph (see Figure 4). Platforms are attributed with parameters for the passenger exchange time distribution, splits are attributed with a typically low local maximum velocity, while track segments are attributed with length, planned traversal times, and maximum velocities that might parametrize local conditions like tight bends, slopes, or pedestrian zones.

Cologne's light-rail network is serviced by vehicles of types *Flexity Swift K4000* (see [27]), *K4500* (see [28]), and *K5000* (see [29]) by Bombardier Transportation. The agents representing these vehicle types are attributed with basic maneuvering attributes, e.g. acceleration and deceleration functions, as well as typical

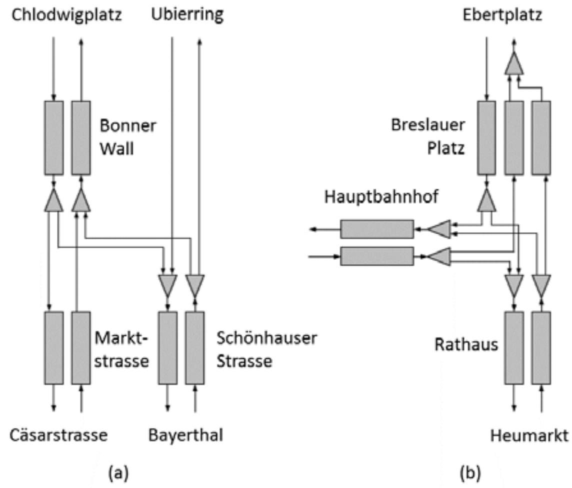


Figure 4: Model representations of the tunnel at its (a) southern and (b) northern end points .

and maximum velocity, passenger capacity, and parameters for the passenger exchange distribution.

2 Experiments

2.1 Scenario and parameters

As the exact timetable to be applied to the reformed routes 5 and 16 has not yet been announced, a discrete optimization model (described in detail in [24], for recent overviews of timetable optimization models see [5] and [6]) combining a genetic algorithm (see [8]) and a branch-and-bound solver is used to generate timetables optimally fitted to the network's characteristics. While the optimization process starts out with an initial population M_i of timetables that are typically not well adapted to the network, timetables from the set of optimum timetables M_o show maximal regularity and best possible adherence to basic transport planning requirements at the same time. Contrasting simulation results for these two sets of timetables allows to examine the impact of regular timetabling on Cologne's new light-rail tunnel, and by extension the impact of the tunnel on overall network behavior.

The optimization model uses regularity of scheduled time offsets between two consecutively departing vehicles at a platform as an indicator for a timetable's robustness against disturbances resulting from small, inevitable delays. For example, in an assumed interval of ten minutes two lines could be scheduled with equidistant offsets of five minutes, which means that vehicles of one or both involved lines could be late for more than four minutes without consequences for vehicles of the following line. Under an extremely unequal split of the available time span into a nine minute offset followed by a one minute offset, vehicles of the first line could have a delay of more than eight minutes without consequences to vehicles of the following line. On the other hand, would the vehicle of the second line be even slightly late, the delay would spread to the follow-up tram. Since we assume typically small operational delays, we see a regular offset distribution as very robust, the occurrence of very small offsets as not robust (see [26]).

Starting out from an objective function value of 191.19 the optimizer yields a plateau of 282,000 optimum solutions with a objective function value of 174.64, an improvement of nominally 8.7 percent. For a first set of experiments ten timetables each are selected randomly from the pool of initial candidates M_i and from the pool of optimum solutions M_o. For each of these timetables ten runs simulating typical operational days are executed. For a more specific look on areas where calculating averages between different timetables does not yield real insight two timetables $\mu_i \in M_i$ and $\mu_o \in M_o$ are selected. For each of these two timetables 100 simulation runs are executed.

For each of these simulation runs a moderate probability of operational delays $p_d = 0.3$, and a moderate distribution of passenger loading times with $a_{h,v} = 0$, $b_{h,v} = 30$, $c_{h,v} = 15$, and a minimum of $m_v = 12$ is assumed (for a detailed discussion of these parameters see [14]).

2.2 Results and discussion

Averaging over all stops in the network and all timetables in M_i, the simulation runs yield an average delay of departures of 20.0 seconds, with a reduction of 3.2 seconds or 16.0 percent to 16.8 seconds under the timetables in M_o. This behavior is consistent with observations made of the Cologne network in its state before the tunnel's completion, which depict a reduction from 19.4 by 3.4 seconds or 14.4 percent to 16.0 seconds (see [23], pp. 156-174). Disregarding punctual departures the average delay is reduced from 36.6 down to 31.4 seconds, a reduction of 5.2 seconds or 14.3 percent. This again is comparable with a reduction from 36.8 by 5.3 seconds or 14.4 percent to 31.5 in the pre-tunnel network.

As described, as part of the planned service line 16

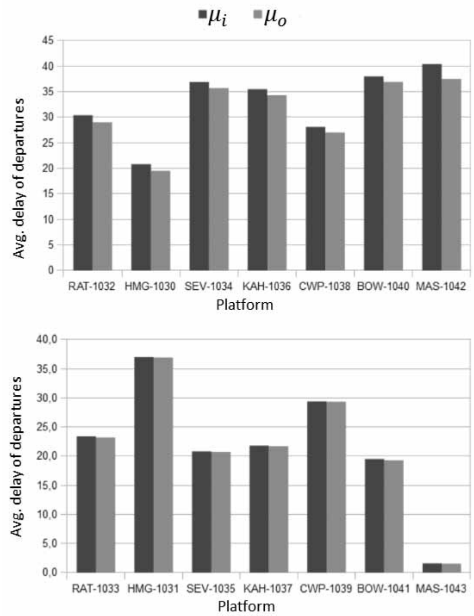


Figure 5: Average delays of departures at stations in or near the light-rail tunnel: southbound (top) and northbound (bottom).

will be rerouted through the new north-south tunnel. The stations in the existing east-west tunnel – which it currently traverses – will be served by one line less, allowing for larger intervals between individual vehicles. Accordingly, departures in the existing tunnel show an average delay of 3.3 seconds for the planned service, independent of the examined schedule. This is a slight reduction in comparison to average pre-tunnel delays of 6.5 seconds under initial timetables, and 4.4 seconds under optimum timetables.

More interesting than these general indicators is a closer view of the light-rail stations in the tunnel and at its entries and exits: Averaged over all tunnel platforms the departure delay is reduced slightly from 21.2 seconds under timetable μ_i to 19.5 seconds under μ_o, a decrease of 8.0 percent or 1.7 seconds. While the departures at platforms oriented southbound show a slight delay reduction, the northbound platforms show no significant change (see Figure 5). With the average delay barely changing under different timetables, the relatively high (with exception of the northbound tunnel entry point Marktstrasse, MAS-1043) basic delay values from 19.3 to 37.5 seconds have to be dependent on other factors. To examine this situation closer, the tunnel-traversing line routes 5 and 16 are discussed.

The delay development of southbound routes 5 and 16 (see Figure 6 (a) and (b)) share central characteristics: Both routes show increases in delay of approximately 16 to 18 seconds between Heumarkt (HMG) and Severinsstrasse (SEV), and of approximately 20 seconds between Chlowigplatz (CHW) and Marktstrasse (MAS). In addition, line 16 has a relatively high delay plateau of approximately 53 seconds when entering the tunnel, which can be explained by the necessity to navigate the highly loaded switch clusters between Reichensperger Platz (RPP), Ebertplatz (EBP), and Breslauer Platz (BRE). It regains some punctuality (approximately 24 seconds) between stations Breslauer Platz and Heumarkt.

For the northbound routes 5 and 16 (see Figure 6 (c) and (d)) the simulation shows delay increases of approximately 16 seconds between Severinsstrasse and Heumarkt, as well as gained punctuality of approximately 29 seconds between Heumarkt and Hauptbahnhof (DOM), and 36 seconds from Heumarkt to Breslauer Platz, respectively. Route 16 shows a relatively high delay of approximately 32 seconds when entering the tunnel at Schönhauser Strasse (SHS).

The simulation demonstrates that the transit vehicles leave the tunnel with approximately the same delay they have when entering it, without displaying significant differences in amount and development of delays under different timetables. The delays are therefore not dependent on the applied schedule, but on other bottlenecks: the switch clusters with their low maximum velocity, and the trains' acceleration and deceleration capabilities that are not adequate to counterbalance delays developing on the relatively short track segments.

Increasing the scheduled traversal time between Reichensperger Platz and Ebertplatz for southbound trains, and before Schönhauser Strasse for northbound vehicles would eliminate most delays. This could be at least partially compensated by reducing the planned traversal time between Heumarkt and Breslauer Platz. Alternatively, using vehicles with higher typical acceleration and deceleration capacities would yield a higher average velocity on the short track segments between tunnel stations.

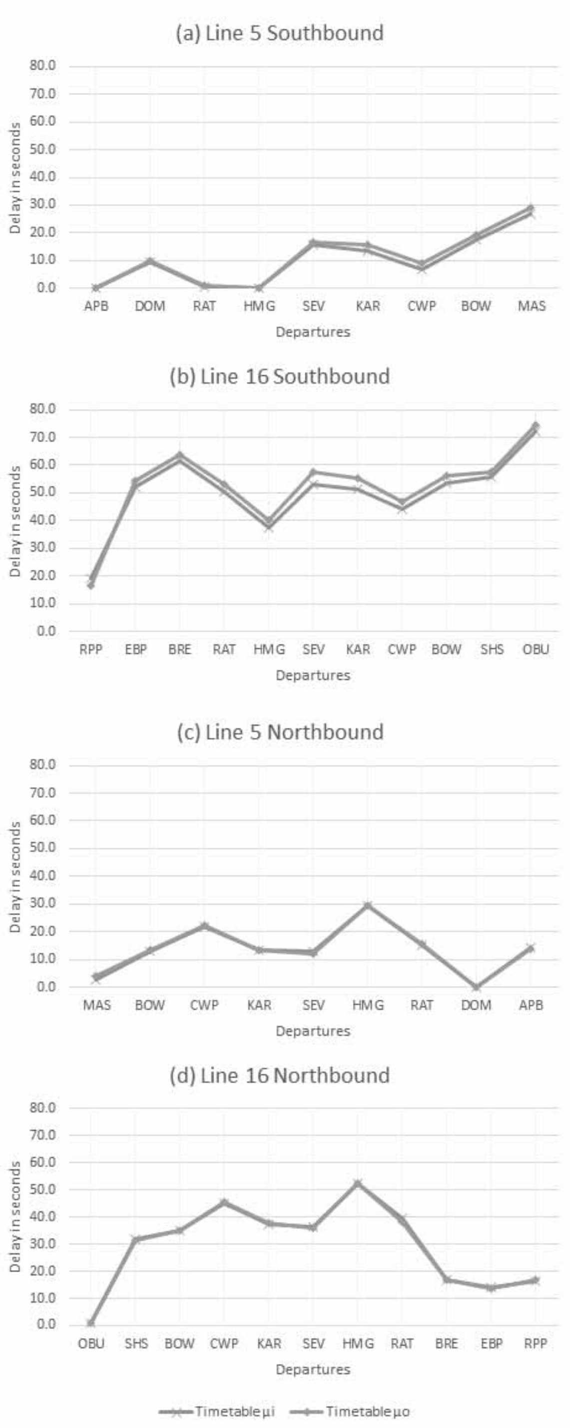


Figure 6: Average delays at departures for line routes 5 and 16 in or near the subway tunnel: southbound ((a) and (b)) and northbound ((c) and (d)).

3 Conclusions

This paper examined the planned subway tunnel connecting Cologne's central station with the central and southern urban districts. After describing characteristics of the tunnel, and the integration of the tunnel and the re-routed subway lines with a simulation model of Cologne's light-rail transit network, some experiments were conducted to both estimate the tunnel's impact on system performance, and the to be expected delays originating from small operational disturbances.

The experiments showed that no overall negative impact on the transit system's performance would have to be expected from the tunnel's integration. While the simulation predicts significant delays at platforms in and around the tunnel, these could be mitigated by increasing the planned traversal time by one minute, in particular in the vicinity of the southbound platform Ebertplatz and the northbound platform Schönhauser Strasse. Deploying transit vehicles with improved acceleration and deceleration capabilities also would increase the tram's average velocity and reduce the observed delays.

4 Acknowledgments

The authors thank Benjamin Brandt for providing the maps depicting course and details of the subway tunnel. 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. Includes material licensed by TerraFly (http://terrafly.com) and the NSF CAKE Center (http://cake.fiu.edu).

References

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

[2] Bartsch M, Brandt A, Kleinhubbert G, Schreiber M. Were Subway Builders Cautious Enough? In: Der Spiegel, March 9, 2009. URL: http://www.spiegel.de/international/germany/cologne-archive-catastrophe-were-subway-builders-cautious-enough-a-612129.html, accessed on April 20, 2017.

[3] Berger P. Bei Tempo 20 vibriert es weniger. In: Koelner Stadtanzeiger, January 10, 2013. URL: http://www.ksta.de/koeln/u-bahn-tunnel-bei-tempo-20-vibriert-es-weniger-3970740, accessed on April 20, 2017.

[4] Borgers M. Karneval im Schatten eines Bauskandals. In: Deutsche Welle, February 13, 2010. URL: http://www.dw.com/de/karneval-im-schatten-eines-bauskandals/a-5245340, accessed on April 20, 2017.

[5] Cacchiani V, Toth P. Nominal and robust train timetabling problems. In: European Journal of Operational Research, Vol. 219, 2012, pp. 727-737.

[6] Caimi G, Kroon L, Liebchen C. Models for railway timetable optimization: Applicability and applications in practice. In: Journal of Rail Transport Planning & Management, Vol. 6, No. 4, 2017, pp. 285-312.

[7] Cats O. Multi-agent Transit Operations and Assignment Model. In: Procedia Computer Science, Vol. 19, 2013, pp. 809-814.

[8] Dreo J, Petrowski A, Siarry P, Taillard E. Metaheuristics for Hard Optimization. Springer, 2006.

[9] Kendziorra A, Weber M. Extensions for logistic and public transport in SUMO. In: SUMO 2015 - Intermodal Simulation for Intermodal Transport, Berichte aus dem DLR-Institut für Verkehrstechnik, Vol. 28, 2015, pp. 83-90.

[10] Kerstholt M. Und noch immer ist niemand schuld. In: tagesschau.de, March 3, 2017. URL: http://www.tagesschau.de/inland/achtjahre-stadtarchiv-koeln-101.html, accessed on April 20, 2017.

[11] Krings D. Das zweite Opfer des Kölner Tunnelbaus. In: Die Welt, November 21, 2004. URL: https://www.welt.de/vermischtes/article118257/Das-zweite-Opfer-des-Koelner-Tunnelbaus.html, accessed on April 20, 2017.

[12] Krischke W. Nord-Süd Stadtbahn Köln – Tunnelbau in der City. Kölner Verkehrsbetriebe AG (Ed.), 2008. URL: http://www.nord-sued-stadtbahn.de/downloadRepository/tunnelbau_in_der_city.pdf, accessed on April 20, 2017.

[13] Lorenz S, Hendorf M. Vermutlich Baufehler – Staatsanwaltschaft bereitet Klage vor. In: Kölnische Rundschau, February 2, 2017. URL: http://www.rundschau-online.de/region/koeln/koelner-stadtarchiv-vermutlich-baufehler—staatsanwaltschaft-bereitet-klage-vor-25661364, accessed on May 1, 2017.

[14] Lückemeyer G. A Traffic Simulation System Increasing the Efficiency of Schedule Design for Public Transport Systems Based on Scarce Data. Aachen, Shaker Verlag, Dissertation, 2007.

[15] Lückerath D, Ullrich O, Speckenmeyer E. Modeling time table based tram traffic. In: Simulation Notes

Europe (SNE), Volume 22, Number 2, August 2012, pp. 61-68.

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

[17] Middelkoop D, Bouwman M. SIMONE: Large Scale Train Network Simulations. In: B.A. Peters, J.S. Smith, D.J. Medeiros, M.W. Rohrer (Ed.): Proceedings of the 2011 Winter Simulation Conference, Arlington, 2011.

[18] Moeck T. Gasalarm legt Kölner Innenstadt lahm. In: Kölner Stadtanzeiger, August 16, 2007. URL: http://www.ksta.de/gasalarm-legt-koelner-innenstadt-lahm-13610876, accessed on April 20, 2017.

[19] Müller S. Kölner U-Bahn-Bau – Jedes zweite Haus ist geschädigt. In: Kölner Stadtanzeiger, April 16, 2009. URL: http://www.ksta.de/koelner-u-bahn-bau-jedes-zweite-haus-ist-geschaedigt-12796062, accessed on April 20, 2017.

[20] Nash A, Huerlimann D. Railroad Simulation Using OpenTrack. In: Allan, J., R.J. Hill, C.A. Brebbia, G. Sciutto, S. Sone (Ed.): Computers in Railways IX, WIT Press, Southampton, 2004, pp. 45-54.

[21] Rossmann A. Der Preis der U-Bahn. In: Frankfurter Allgemeine Zeitung, March 5, 2009. URL: http://www.faz.net/aktuell/feuilleton/kunst/koelner-stadtarchiv-der-preis-der-u-bahn-1924780.html, accessed on April 20, 2017.

[22] Suzumura T, McArdle G, Kanezashi H. A high performance multi-modal traffic simulation platform and its case study with the Dublin city. In: Proceedings of the 2015 Winter Simulation Conference, 2015, pp. 767-778.

[23] Ullrich O. Modellbasierte Parallelisierung von Anwendungen zur Verkehrssimulation - Ein dynamischer und adaptiver Ansatz. Dissertation, Univ. Köln, 2014.

[24] Ullrich O, Lückerath D, Franz S, Speckenmeyer E. Simulation and optimization of Cologne's tram schedule. In: Simulation Notes Europe (SNE), Volume 22, Number 2, August 2012, pp. 69-76.

[25] Ullrich O, Lückerath D. An Introduction to Discrete-Event Modeling and Simulation. Submitted to Simulation Notes Europe (SNE).

[26] Ullrich O, Lückerath 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.

[27] Vossloh Kiepe GmbH: Elektrische Ausrüstung der Niederflur-Stadtbahnwagen K4000 der Kölner Verkehrs-Betriebe AG. (Druckschrift Nr. 00KV7DE), 2003.

[28] Vossloh Kiepe GmbH: Elektrische Ausrüstung des Niederflur-Stadtbahnwagens K4500 für die Kölner Verkehrs-Betriebe AG (Druckschrift Nr. 00KN2DE), 2003.

[29] Vossloh Kiepe GmbH: Elektrische Ausrüstung der Hochflur-Stadtbahnwagen K5000 der Kölner Verkehrs-Betriebe AG (Druckschrift Nr. 00KB5DE), 2003.

Experimental Adaptation of a Training Simulator for Manual Welding Processes towards the Teach-In of Welding Robots

Benjamin Knoke[1*], Mareike Voskuhl[2], Marcel Tebbe[2], Markus Häveker[3], Christian Gorldt[1], Klaus-Dieter Thoben[1]

[1]BIBA – Bremer Institut für Produktion und Logistik GmbH, University of Bremen, Hochschulring 20, 28359 Bremen, Germany; **kno@biba.uni-bremen.de*

[2]University of Bremen, Bibliothekstraße 1, 28359 Bremen, Germany

[3]Daimler AG, Mercedesstraße 1, 28309 Bremen

SNE 28(1), 2018, 27 - 30, DOI: 10.11128/sne.28.en.10404
Received: March 10, 2018 (Selected ASIM GMMS/STS 2018 Conf. Publ.), Accepted: March 15, 2018
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. This paper describes an attempt to provide assistance during the teach-in process of welding robots by an experimental adaptation of training simulator components. A mounted camera system continuously monitors the relative position and orientation of the welding torch and workpiece. The simulation contains images of metal textures and a welding seam that are computed in real-time and that are displayed using augmented reality technology. Feedback on deviations from the ideal position, orientation, and movement is displayed during the process and can also be analysed afterwards. The experimental application is strongly limited by the restriction to a given set of basic work-pieces. However, the visual feedback was considered by the programmers to be very helpful and the application showed great potential that could be used for dedicated industrial products.

Introduction

Automatisation technology within the automotive sector strongly relies on welding robots [1]. Due to the ongoing trend towards a ‘digital factory’, this is not expected to change within the foreseeable future [2], [3].

The setup of the welding robots is usually performed by a qualified programmer in multiple iterations. The body parts that are used during setup are usually scrapped. Nowadays, the welding programmes are often prepared in CAD environments or during other forms of simulations, but this still does not lead to acceptable quality levels without conventional testing. As current car models inherit up to 600 complex welding seams, this results in considerable costs.

The presented work focuses on the experimental adaptation of a welding simulator, which is usually used within manual welding training exercises, to guide the programming of welding robots. This simulator visually tracks the relative positions and orientations of welding gear and workpiece to continuously provide feedback to the user. This paper investigates the application of the generated feedback information to guide the teach-in process of welding robots.

1 State of the Technology

1.1 Programming of welding robots

The most prevalent methods for the programming of welding robots include the generation of programmes based on CAD models, where the path is defined within a CAD environment [4], and the programming by demonstration technique. The latter is commonly used within the automotive sector [5]. During such teach-in processes, the robots are manually guided along their desired path. This path is saved as a program that can later be accessed to perform the teached tasks within the production line.

In industrial practice, the programming is usually applied by dedicated programmers with limited welding experience. However, some complex welding seams require careful tuning of the welding parameters [6]. This results in one of the major causes for delays during the startup of new production lines.

A similar approach to the presented method has been proposed by Ni et al. (2017). It generates 3D models of scanned workpieces and employs a haptic input device to define the welding path. The programming is supported by an AR interface as well as haptic force feedback. This approach is designed for scenarios in which the users do not have physical in which the users do not have physical access to the workpieces and the feedback is limited to the positioning of the welding seam, not the generated quality [7].

1.2 Training simulators for manual welding

The current key technologies in welding simulations are Virtual Reality and Augmented Reality. During the past decade these technologies were applied to create welding simulators and made the transition towards practical application [8], [9].

These simulators rely on optic measurement to capture the position and movement of a welding torch or electrode [10]. The measured characteristics include stick out, work angle, travel angle, travel path, and travel speed. The available welding simulators differ greatly in physical and functional fidelity, as well as in price.

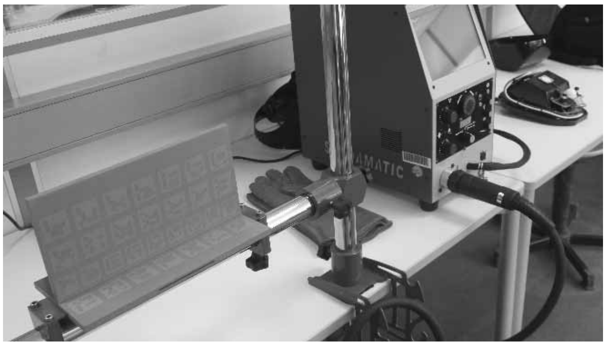


Figure 1. Soldamatic welding simulator.

The applied simulator system is the Soldamatic training simulator for welders (Figure 1), which focusses on manual welding processes and is a commercial product sold by Seabery Soluciones. SL [11]. The hardware consists of a computing unit that is embedded in a welding equipment casing; a welder's helmet that holds cameras, speakers and an integrated display; attachable hand-held devices for MIG/MAG, electrodes, and GTAW welding processes; a stand; and multiple standard work pieces that are made out of plastic and are printed with reference markers [11], [12].

Figure 2. Soldamatic evaluation screen.

A mounted camera system continuously monitors the relative position and orientation of the welding torch and workpiece. The simulation contains images of metal textures and a welding seam that are computed in real-time and that are displayed using augmented reality technology. Feedback on deviations from the ideal position, orientation, and movement is displayed during the process and can also be analysed afterwards (Figure 2).

2 Methodology

The paper describes an attempt to provide an assistance during the teach-in process by experimental adaptation of training simulator components. The experiment uses the MIG/MAG simulation in combination with a T-joint workpiece.

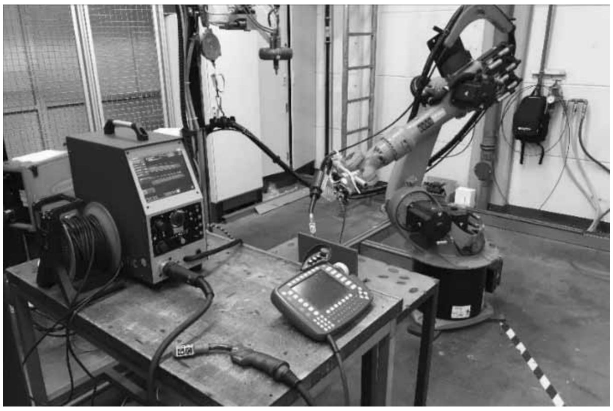


Figure 3. Experimental setup.

The main components are the Soldamatic welding simulator as well as a KUKA KR15 industrial robot with six axes and a payload of 15kg. The complete assembly is depicted in Figure 3. The welding helmet of the simulator has been disassembled, as only the camera system was required. Additionally, a light attachment has been constructed to mount the camera system on the industrial robot (Figure 4). The artificial welding gun for the robot has been created with a 3D printer to resemble the simulator's MIG/MAG gun, as it is required to hold the reference markers in their original position.

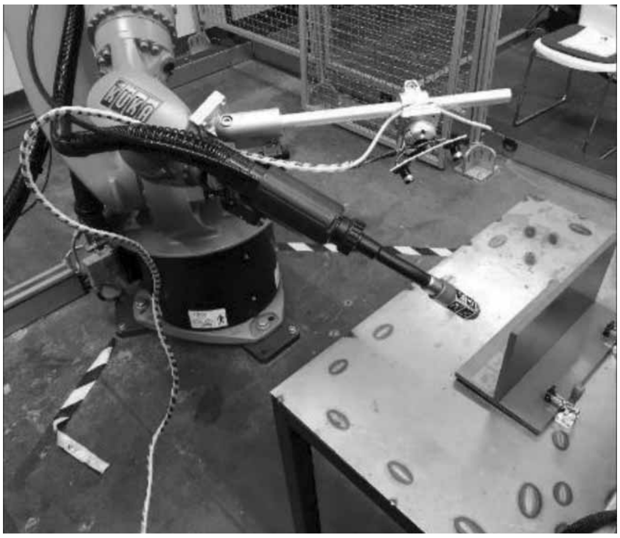


Figure 4. Camera mount and custom welding torch in position over the workpiece.

The simulation parameters are listed in Table 1.

parameter	value
difficulty	beginner
material	steel
workpiece thickness	10mm
welding additive diameter	1mm
seam type	single-layer fillet weld
protective gas	argon mixed gas
current	23A
voltage	30V

Table 1. Selected simulation parameters.

The experiment has been run in a configuration for beginners, as this offers the most guidance during the process. The simulations' feedback has been applied to tune the position, orientation, and movement of the welding robot.

The robot has been programmed with an average speed of 1,5mm/s and five reference points have been defined along the weld. Once the calibration has been optimized according to the simulation, the exercise was performed multiple times.

The final evaluation has been performed by industrial experts, who included welding trainees, welding trainers, and maintenance staff.

3 Findings and Limitations

Once the programming of the welding robot had been tuned according to the feedback of the simulation, the repetition of the experiment constantly led to positive results, which indicates a certain degree of consistency.

The industrial experts stated the application could be used to display and correct position and orientation of the welding torch relative to the workpiece. The ideal configuration of these parameters was considered to be similar to the programming made based on the simulation. However, the velocity that is demanded by the simulation, is estimated to be too low for industrial practice and may ultimately lead to the destruction of the workpiece due to burning through.

The practical usability of the simulations in production was rated as high by the maintenance employee. Even trained welding experts benefit in his opinion from the guidance, since the programming of a welding robot, in contrast to manual welding, cannot be performed intuitively. The technology also provides the opportunity to have an initial teach-in done by a robot expert, so that a welding expert is only required to assist at the end of the teach-in process for additional fine-tuning operations.

It was also highlighted that the working position of the programmer and his vision on the task are improved by the simulation. Only in rare moments, a detailed look at the exact position of the nozzle must be taken. The adjustment of angle and distance can be done on the monitor of the simulation device. This could be particularly helpful in the case of inaccessible or unergonomic seam positions.

Overall, the experts were sceptical as to whether the simulation device can reproduce the very complex processes involved in industrial welds. Necessary changes in the speed with varying wall thicknesses, material changes or elaborate curves are initially not provided in the system.

4 Conclusion and Outlook

The experimental application of the training simulator towards the teach-in of welding robots has been performed for a sample T-joint workpiece. The industrial experts reported the feedback through Augmented Reality a considerable benefit in finding the correct position and orientation of the welding torch. This benefit would be especially relevant in settings with limited access or visibility.

However, the experimental application is strongly limited by restriction to a given set of basic workpieces. The introduction of additional workpieces to the Soldamatic simulation has already been performed in few cases but requires considerable effort and interaction with the developer. The approach described by Ni et al. (2017) employs a scanning module that creates point clouds of the workpiece [7] and could be a possible approach to overcome this barrier. A method to define the seam positions would be required as well, which should be flexible enough to follow the outline of complex seams and allow for changes in material thickness.

Additional challenges were imposed by the optical sensors, which had to be positioned at an average distance of 300 mm to the workpiece. The recognition was impeded by vibrations of the system that were caused by robot movement. A dedicated application should be designed rather compact to allow for welds that are difficult to access.

Acknowledgement

The authors would like to thank the Mercedes-Benz plant in Bremen as well as the Daimler Zukunftswerkstatt for the opportunity and support necessary to realize the experiments, as well as the German Federal Ministry of Education and Research (BMBF) for their support in the project 'MESA - Media Use in Welding Education' (grant number 01PD14016).

References

[1] Bauernhansl T, Ten Hompel M, Vogel-Heuser B. *Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung, Technologien und Migration.* Wiesbaden: Springer, Vieweg; 2014.

[2] Hömberg D, Landry C, Skutella M, Welz WA. Automatic Reconfiguration of Robotic Welding Cells. *Mathindustry* 2017; 27: 183-203.

[3] Lüder A, Schmidt N. Challenges of Mechatronical Engineering of Production Systems: An Automation System Engineering View. *Mathindustry* 2017; 27: 93-114.

[4] Ferreira LA, Figueira YL, Iglesias IF, Souto MÁ. Offline CAD-based Robot Programming and Welding Parametrization of a Flexible and Adaptive Robotic Cell Using Enriched CAD/CAM System for Shipbuilding. *Procedia Manufacturing* 2017; 11: 215-223.

[5] Kah P, Shrestha M, Hiltunen E, Martikainen J. Robotic arc welding sensors and programming in industrial applications. *International Journal of Mechanical and Materials Engineering* 2015; 10(13).

[6] Horváth CM, Korondi P, Thomessen T. Robotized multipass Tungsten Inner Gas welding of Francis hydro power turbines. *Proceedings of the 26th International Symposium on Industrial Electronics (ISIE)*; 2017; 1759-1765.

[7] Ni D, Yew AWW, Ong SK, Nee AYC, Haptic and visual augmented reality interface for programming welding robots. *Advances in Manufacturing* 2017; 5(3): 191-198.

[8] Fast K, Gifford T, Yancey R. Virtual training for welding. *Proceedings of the 3rd IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2004)*; 2004; Arlington; 298-299.

[9] White S, Prachyabrued M, Baghi D, Aglawe A, Reiners D, Borst C, Chambers T. Virtual welder trainer. *Proceedings of the IEEE Virtual Reality Conference 2009*; 2009; Lafayette; 303-303.

[10] Okimoto MLL, Okimoto PC, Goldbach CE. User experience in augmented reality applied to the welding education. *Procedia Manufacturing*; 2015; 3: 6223-6227.

[11] Knoke B, Thoben KD. Integration of Simulation-based Training for Welders. *Simulation Notes Europe* 2017; 27(1): 37-44.

[12] Seabery Soluciones. Soldamatic educational augmented reality: la tecnología educativa más avanzada y competitiva para la formación de soldadores, 100% española. *Soldadura y tecnologías de union* 2012; 23(130): 58-60.

A Comparative Solution to ARGESIM Benchmark C4 'Dining Philosophers' with AnyLogic and MVSTUDIUM

Joshua Grove

Institute for Distributed Intelligent Systems, Saint-Petersburg State Polytechnical University, Saint-Petersburg, *Russia; jgrove_2000@yahoo.com*

SNE 28(1), 2018, 31 - 33, DOI: 10.11128/sne.28.bn04.10406
Received: February 20, 2012;
Revised by EO August 20, 2017; Accepted: October 10, 2017
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna,
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. ARGESIM Benchmark C4 Dining Philosophers' is a very general one, and various modelling approaches are suitable. The presented solution compares modelling approaches with two similar simulators, but different in goals: AnyLogic and MVSTUDIUM. After a short introduction into the two object-oriented simulators, event-driven hybrid modelling approaches are presented – here the solution refers to the noticeable difference between the two modeling environments in messaging and signal processing. As option – and in order to üprevent from deadlock – also a chopstick cleaning process is introduced, and the AnyLogic version provides a 2-D animation directly defined in the philosophers' objects. Finally some results for utilization and waiting times are presented – identically for both simulators, because of use of identical random streams.

1 Simulators

AnyLogic is an integrated graphical modeling environment from XJ Technologies. It is strongly Java based and allows for custom code, external libraries, and external data sources. Version 6.4.1 was used for this model. A 15-day demo trial with 40 different categories ofexamples can be downloaded from their website (http://www.xjtek.com/).

MVSTUDIUM is an integrated graphical modeling environment from the MVSTUDIUM Group. MVSTUDIUM is capable of modeling multi-component systems with continuous and hybrid processes. It is object oriented and based on UML (Unified Modeling Language) notation.

MVSTUDIUM does not require any traditional programming to build models. It allows for graphical programming and entering systems of second order differential, algebraic, and differential-algebraic equations. MVSTUDIUM includes wizards for easy 2D and 3D-animation. Version 4.2 was used for this model. A 30-day demo trial version with ten working examples can be downloaded from the MVSTUDIUM website (www.mvstudium.com).

2 Modelling

In MVSTUDIUM, an object oriented event driven approach was taken for modeling the *Dinning Philosophers*. Two interacting hybrid classes were created. They were a table class and a philosopher class. In a similar way, two *Active Object Classes* were created in AnyLogic. The philosophers send requests to the table for sticks and the table responds by signaling if the request was successful or not.

A noticeable difference between the two modeling environments is the messaging and signal processing. AnyLogic allows for bi-directional connections between objects, where as a separate connection is required for MVSTUDIUM. In AnyLogic, sending messages with multiple parameters requires the programming of Java classes and adding code to connect the message to the state chart.

The *Table Class* created for the model contains a vector that indicates which of the five sticks are present or absent on the table at any time. Figure 1 and Figure 2 show the behavior chart, which is based on the UML state diagram, of a philosopher with the deadlock condition in AnyLogic and MVSTUDIUM, respectively.

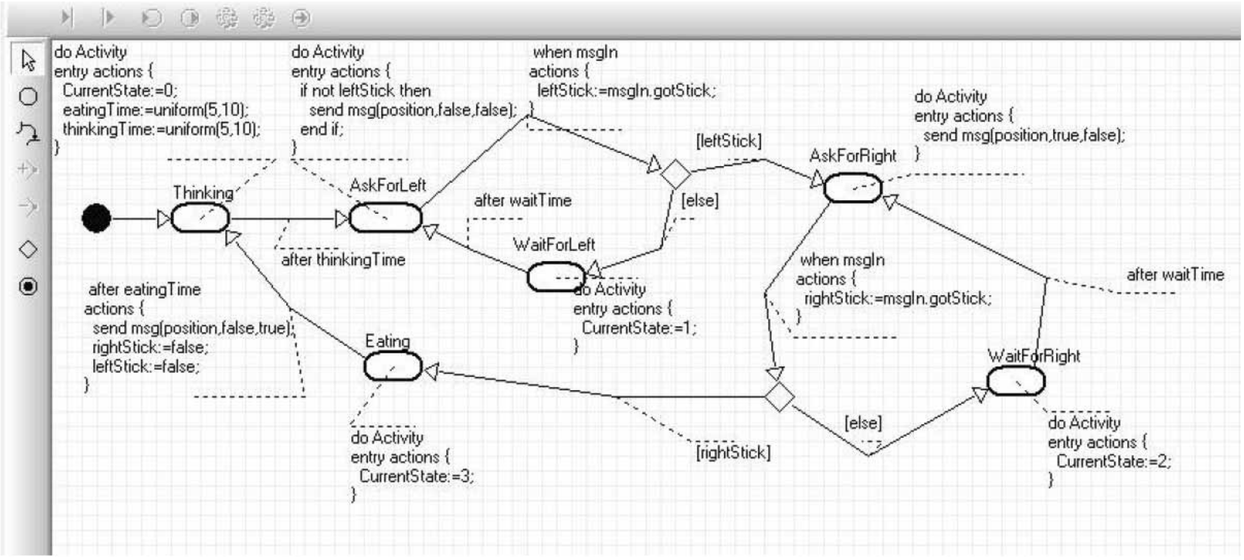


Figure 2. Behaviour Chart in MVSTUDIUM

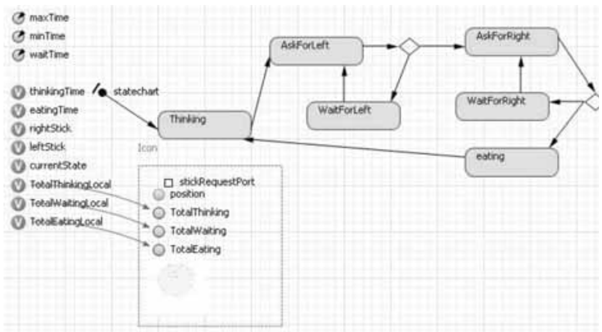


Figure 1. Behaviour Chart in AnyLogic.

The philosophers may starve or enter a deadlock condition, if they all grab a left stick before a philosopher can obtain a right stick. By adjusting the eating and thinking times a deadlock can become more probable. For instance a deadlock is more likely with a eating and thinking time at randomly selected in the interval of 0.5 to 1.5 than from an interval of 1 to 10 seconds.

For this comparison, three solution strategies were implemented in both modeling software solutions. They included

- a waiter that prevented utilization of all the chopsticks,
- a cleaning process that allowed communication between philosophers,
- and a put back strategy.

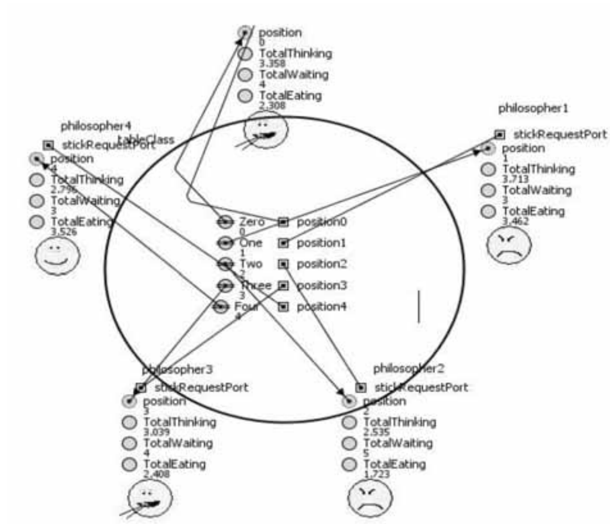


Figure 3. 2-D animation in AnyLogic.

In AnyLogic, a 2-D animation was created directly in the philosopher class presentation to illustrate the philosopher's state (Figure 3).

In MVSTUDIUM, a similar animation was created in the testbench for each philosopher (Figure 5).

An integer value was used by both modeling software tools to indicate the state of the philosopher (thinking, waiting for sticks, waiting for one stick, and eating) and to switch the animation to the appropriate image (Figure 4).

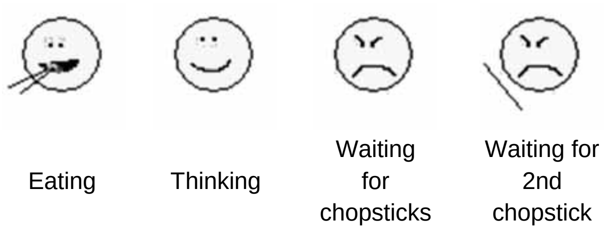


Figure 4. Status of philosophers in animations.

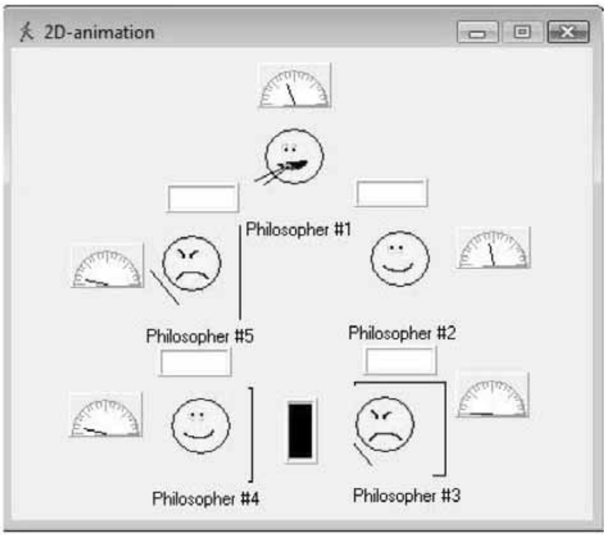


Figure 5. 2-D animation in MVSTUDIUM.

When creating the visual representation, the ability to create animations directly in the class in AnyLogic reduced the duplication of work by not having to create an animation for every instance of the philosopher class as required in MVSTUDIUM.

However, MVSTUDIUM 2D animation creator has a drag and drop style which is very easy to use.

3 Results

Table 1 shows the average times of the five philosophers in the waiting (hungry), thinking, and eating states for each of the different scenarios. The time for each simulation run was 100 time units.

Eating and thinking times were set randomly to be values between 0.5 and 1.5 time units. A waiting time of one time unit was used before the philosopher tried to obtain a stick after a failed attempt to obtain a stick from the table.

A deadlock was quickly achieved in the case with no strategy and the philosophers spent almost the whole time starving. The Put-back strategy achieved the longest time thinking and eating, followed by the waiter, and cleaning strategies.

The random eating and thinking time intervals were saved in a matrix and used for both AnyLogic and MVSTUDIUM. Subsequently, identical results were achieved from both.

	Deadlock	Waiter	Cleaning	Put back
Waiting	92.8	53.4	60.2	46.0
Thinking	3.7	23.1	19.8	27.0
Eating	2.9	22.8	19.7	26.7

Table 1. 2-Results AnyLogic and MVSTUDIUM.

An Object-oriented Approach to ARGESIM Benchmark C14 'Supply Chain' using MATLAB

Matthias Wastian[1*], Stephan Reichl[2]

[1]dwh Simulation Services, Neustiftgasse 57-59, 1070, Vienna; *Matthias.Wastian@dwh.at*

[2]MMS- Mathematical Modelling and Simulation, Inst. of Analysis and Scientific Computing, TU Wien, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria;

SNE 28(1), 2018, 35 - 38, DOI: 10.11128/sne.28.bn14.10407
Received: August 10, 2016; Revised August 15, 2017;
Accepted: December 10, 2017
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna,
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. ARGESIM Benchmark C14 'Supply Chain Management' allows different modelling approaches, from classical simulation approaches with discrete event systems or process modelling to directly programmed system evaluation. The tasks require a classical feedforward planning mechanism, without implicit feedback loops. An intrinsic property of the supply chain is a bi-directional flow, a material flow from factory via distributor to wholesaler, and an order flow from wholesaler via distributor to factory. Some simulation systems provide special modules for supply chain integrating these bidirectional flows, otherwise 'twin' modules must be defined, with reverse flow and intracommunication. Directly programmed modules for the flows may be an efficient approach, but the model flow descriptions are hard to read and not really suitable for understanding of the process. This solution tries an alternative approach using a fully object-oriented modelling approach for the modules fatory, distributor, and wholesaler – defined in MATLAB in order to make use of efficient vector and matrix structures.

1 Simulator

MATLAB – **MAT**rix **LAB**oratory – is a commercial platform independent software by *MathWorks Inc.* and used for solving mathematical problems, rather numerical calculations in the context of matrices. The syntax of MATLAB is adapted to common object oriented programming languages like JAVA or C#.

Hence, it is easy for programmers, who are used to common object oriented languages, to get familiar with the syntax.

MathWorks introduced class and method structures, combined with property definitions. However, MATLAB must not be mistaken with these programming languages. The underlying concept is different and designed for performing matrix calculations. The simulator program has to be implemented by capitalizing from these advantages.

2 Modelling

According to task assignments a supply chain management problem has to be simulated. There are three different types of active participants who are categorized to factories, distributors, and wholesalers.

While the number of the first two groups is fixed to four in each case, the number of wholesalers is summarized in a group. Figure 1 shows the simulation setup, to be mapped into object-oriented MATLAB structures.

Classes are defined for factories, distributors and wholesalers containing properties for matrices representing the stock as well as ordering status. The general structure is:

```
classdef Factory < handle
% FACTORY
   % - constructor
   % - order(factory object, and the curren order)
   % - produce (factory object and the current
             simulation tic)
   % - proof (factory object and current order)
   % - storage (factory object, MAT on
       finished products)
% getter/setter methods
```

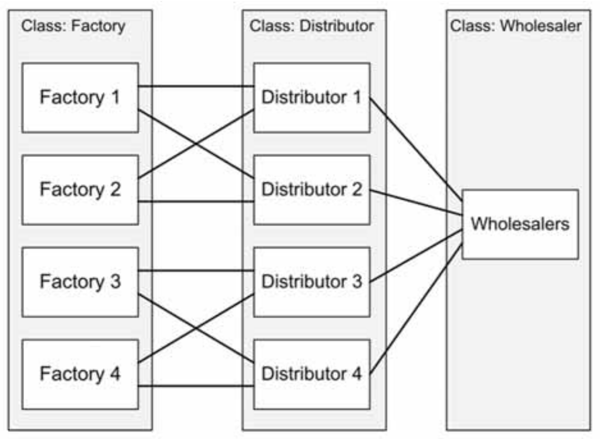


Figure 1. Simulation setup for chain supply with classes.

In general the simulation is tried to be implemented by considering the matrix theorem of MATLAB in order to get higher performance.

In order to boost the simulation speed, the simulation setup is utilized. It defines intervals of seconds which are multiples of 100. Hence these intervals are discretized to intervals of 100 seconds. This method reduces the simulation time by accepting slight changes in the randomized results which can be disregarded.

The basic supply methodology is the following. Factories produce day and night 12 types of products. Each factory produces only six products of the assortment. The choice for the produced product as well as the inter-arrival time at the distributor randomly distributed (uniform and exponential). In the simulation the inter-arrival time is calculated and after is passed the distributor's storage gets filled.

The distributors have specific factories assigned which they order goods from. The orders are placed at the beginning of the day. The group of wholesalers order from the distributors. The order time and the selected distributor are randomly chosen (uniformly distributed). The ordering mechanism is varied to three types `Simple`, `Demand`, `Order-Del`ay ordering mechanisms.

The production matrix defines the products which are produced by certain factories:

```
ProdMAT=
[[1,1,1,1,1,1,0,0,0,0,0,0]; [0,0,0,0,0,0,1,1,1,1,1,1];
 [0,0,0,1,1,1,1,1,1,0,0,0]; [1,1,1,0,0,0,0,0,0,1,1,1]];
```

The production matrix is allocated to a specific factory instance:

```
FA=Factory(ProdMAT(1,:), 'A');
```

The distributors instances are initialized with the certain run initial values like, time of distribution, the factory matrix where goods can be ordered and the ordering strategy:

```
DistributorA= Distributors([16,22,20,12],
              FactoryMAT, 'A', OSTRATEGY);
```

The wholesaler instance receive the distributor matrix on the distributors:

```
Wholesaler(DistributorMAT);
```

3 Results Task a – Simple Order Strategy

In Task a – Simple Order Strategy - the simple ordering mechanism is used in between the distributor and the factories after the preordering – 10 pieces per product. If the distributor sells at least one piece of a product he orders two pieces of the same product.

In Figure 2 the evolution of distributor 1's stock is shown during 30 days. On the seventh day the distributors start to order 10 pieces of every product which arrive until day 8. Hence, distributor 1 has a stock of 120 products. On day 9 the wholesalers start to buy products which result in a reduction of the stored products. This is marked by the red line in Figure 2. Afterwards the stock increases due to the ordering strategy.

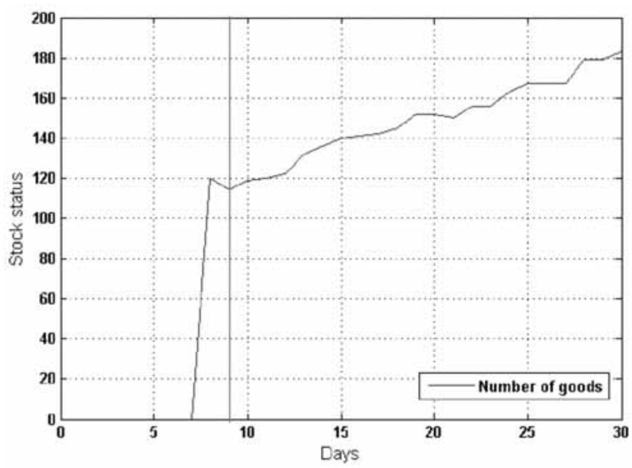


Figure 2. Stock status evolution for distributor 1 - Task a – Simple Order Strategy.

The following Matlab code above shows the procedural calls of the functions for this task:

```
for
 DAYSEC=1+((DAY1)*DAYLENGTH):1:DAYLENGTH*DAY
 factoryProduction(FA,FB,FC,FD,DAYSEC);
 factoryOrder(DistributorA,DistributorB,
         DistributorC, DistributorD,DAYSEC);
 wholesalerOrder(WholesalerG, DAYSEC);
end
```

First the factory production is triggered that starts the day production. Second, the distributors submit their order of the day. Last, the wholesalers send their orders. FA, FB, FC, and FD represent the product matrix for each factory.

The distributor orders a constant number of products without regarding the real requirements. Hence he does not sell as many products as he orders which results in an increasing stock status. As it is shown in Table 3 the simple ordering strategy is the most expensive one. Table 1 lists the maximum and minimum values of the total cost C, the number of delivered products N, and the relative costs R.

	C	N	R
max	12032	246	58.90
min	11141	199	45.24

Table 1. Min and max values of total costs C, delivered products N, and relative costs R for task a – simple order strategy.

The following MATLAB code above shows the `orderStrategyA` represented in source code:

```
function orderStrategyA(this)
   this.currentOrderMAT =
       logical(this.soldMAT).*2;
    this.currentOrderMAT =
      this.currentOrderMAT +
     (sum(this.nextDayOrderMAT,1));
   this.nextDayOrderMAT=zeros(1,12);
   this.soldMAT=zeros(1,12);
end
```

First the current order matrix is calculated. This is done on multiplying the number of sold products with two within the sold matrix. In addition the orders which were not able to be finished the day before are added. Afterwards the matrix is set to a zero matrix.

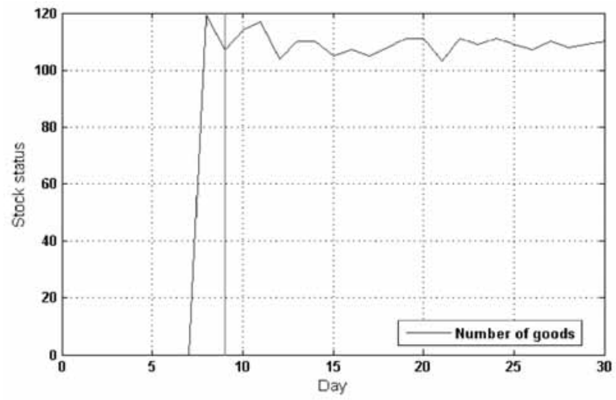


Figure 3. Stock status evolution for distributor 1 - Task b – On Demand Order Strategy.

4 Results Task b – On Demand Order Strategy

In Task b the on demand order strategy is used. The distributor order as many as needed pieces of products every day. Figure 3 shows the stock status over 30 days. Due to the ordering mechanism which is adapted to the actual wholesaler requirements the stock status remains constant.

The MATLAB code below shows the `orderStrategyB` represented in source code.

```
function orderStrategyB(this)
    this.currentOrderMAT = this.soldMAT;
    this.currentOrderMAT =
           this.currentOrderMAT +
           (sum(this.nextDayOrderMAT,1));
this.nextDayOrderMAT = zeros(1,12);
this.soldMAT=zeros(1,12);
end
```

First the current order matrix is calculated. Contrary to Task a, this is done on summing up the sold matrix with the matrix that contains the unfinished order from the day before.

Table 2 shows that the maximum of costs as well as the minimum of costs decreases in comparison with Task a. The number of sold products remains constant.

	C	N	R
max	11031	257	55.90
min	10582	195	41.80

Table 2. Min and max values of total costs C, delivered products N, and relative costs R for Task b – On Demand Order Strategy.

5 Results Task c – Minimal Supply Time - Strategy

The *Order Delay Strategy* in Task c implies the possibility for the distributors to order at the factory with the minimum of inter-arrival time.

Table 3 shows the minimum and maximum values of C, N, and R. Even the stock status is similar to the one in task b – see Figure 4 – and the distributors have the free choice of factories, the costs increase.

That is because of the charging of the costs of delivery which are independent from the number of delivered goods. Hence, the distributor pays the same amount of money regardless of the number of goods delivered.

In Task a and Task b the factories are fixed. One distributor can only order from to specific factories. In Task c it is possible that the distributor orders from more than two factories in one day. This results in increasing delivery costs and directly influences the total costs.

Below, MATLAB code is shown regarding the order strategy of Task c:

```
function orderStrategyC(this);
  calculateStorageCost(this);
  if max(this.currentOrderMAT(1,:))~=0
   contactFactories(this,tic);
end
```

The difference to strategy Task b is the contact of factories for ordering goods. Factories are contacted by getting the factory with the shortest supply time. Therefore the supply-time matrix is sorted:

```
sort(this.supplyTIME)
```

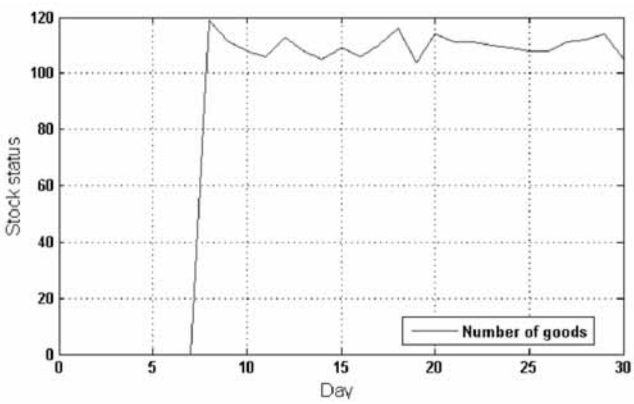


Figure 4. Stock status evolution for distributor 1 - Task c – Minimal Supply Time - Strategy.

	C	N	R
max	11031	257	55.90
min	10582	195	41.80

Table 3. Min and max values of total costs C, delivered products N, and relative costs R for Task c – Minimal Supply Time - Strategy.

6 Summary

The simulation is accomplished in MATLAB. The simulation deals with the advantages and disadvantages of the *Simple Order Strategy*, *On Demand Order Strategy*, and *Minimal Supply Time - Strategy*. These tasks imply a high number of matrices. Hence, the application MATLAB is of advantage as it shows high performance in matrix calculations.

Table 4 shows a comparison of the mean and deviation values of total cost C, the number of delivered products N, and the relative costs R between the three tasks. It is shown that the *On Demand Order Strategy* is the cheapest one. Contrary to the *Simple Order Strategy*, products are ordered on demand which results in lower stock costs. On the other hand the delivery costs are lower than those achieved with the *Minimal Supply Time - Strategy* as the number of factories are limited. Even the delivery costs are limited in Task c by choosing the factory with the lowest delivery time. Hence the total costs get higher than in Task b.

	Task a Mean/ St.Dev	Task b Mean/ St.Dev	Task c Mean/ St.Dev
C	11357 / 1634.5	10482 / 1949	11983 / 2248
N	220.9 / 33.6	218.6 /43.2	217.9 /44.1
R	51.6 / 2.86	48.1 / 3.16	55.3 / 4.4

Table 4. Comparison between order strategies in the three tasks.

SNE Simulation News

EUROSIM Data and Quick Info

EUROSIM 2019

10th EUROSIM Congress on Modelling and Simulation

La Rioja, Logroño, Spain, July 1 – 5, 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 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 Hauptstrasse 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, July 1 – 5, 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 (TU Wien)
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, *thorsten.pawletta@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*
	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, *thorsten.pawletta@hs-wismar.de*
	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
Web EUROSIM	A. Körner, *andreas.koerner@tuwien.ac.at*

Last data update December 2017

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

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

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

ASIM Working Committee	
GMMS	Methods in Modelling and Simulation Th. Pawletta, *thorsten.pawletta@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@dwh.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. 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 Jiménez, *emilio.jimenez@unirioja.es*
Edit. Board SNE	Juan Ignacio Latorre, *juanignacio.latorre@unavarra.es*
Web EUROSIM	Mercedes Perez *mercedes.perez@unirioja.es*

Last data update February 2018

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 December 2012

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 December 2012

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	M. Mujica Mota, *m.mujica.mota@hva.nl*
Vice president	A. Heemink, *a.w.heemink@its.tudelft.nl*
Treasurer	A. Heemink, *a.w.heemink@its.tudelft.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 February 2018

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 December 2012

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 December 2010

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 Massei, *massei@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.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 five Nordic countries Denmark, Finland, Iceland, 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), **MoSis** (Society for Modelling and Simulation in Sweden), **DKSIM** (Dansk Simuleringsforening) and **NFA** (Norsk Forening for Automatisering).

→ *www.scansims.org*
erik.dahlquist@mdh.se
✉ 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	Bernt Lie, *Bernt.Lie@usn.no*
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 February 2018

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	R. Karba, *rihard.karba@fe.uni-lj.si*
Web EUROSIM	Vito Logar, *vito.logar@fe.uni-lj.si*

Last data update February 2018

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.

→*uksim.info*
✉ *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*, *senyb@mail.ru*

Last data update February 2018

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 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

ASIM - Buchreihen / ASIM Book Series

Proceedings

Simulation in Production and Logistics 2017 – 17. ASIM Fachtagung Simulation in Produktion und Logistik
Sigrid Wenzel, Tim Peter (Hrsg.); kassel university press GmbH, Kassel, 2017; ISBN 978-3-7376-0192-4 (print), ISBN 978-3-7376-0193-1 (e-book); ASIM Mitteilung AM164.

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.

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.

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.

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.

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.

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.

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

Patrick Einzinger: **A Comparative Analysis of System Dynamics and Agent-Based Modelling for Health Care Reimbursement Systems.** FBS 24; ISBN 978-3-901608-75-9, ASIM/ARGESIM Vienna, 2014; www.tuverlag.at – print-on-demand.

Martin Bruckner: **Agentenbasierte Simulation von Personenströmen mit unterschiedlichen Charakteristiken.** FBS 23; ISBN 978-3-901608-74-2 (ebook), ASIM/ARGESIM Vienna, 2014; www.tuverlag.at – print-on-demand

Stefan Emrich: **Deployment of Mathematical Simulation Models for Space Management.** FBS 22; ISBN 978-3-901608-73-5 (ebook), ASIM/ARGESIM Vienna, 2013; www.tuverlag.at – print-on-demand

Xenia Descovich: **Lattice Boltzmann Modeling and Simulation of Incompressible Flows in Distensible Tubes for Applications in Hemodynamics.** FBS 21; ISBN 978-3-901608-71-1 (ebook), ASIM/ARGESIM Vienna, 2012; www.tuverlag.at – print-on-demand

Florian Miksch: **Mathematical Modeling for New Insights into Epidemics by Herd Immunity and Serotype Shift.** FBS 20, ISBN 978-3-901608-70-4 (ebook), ASIM/ARGESIM Vienna, 2012; www.tuverlag.at – print-on-demand

Shabnam Tauböck: **Integration of Agent Based Modelling in DEVS for Utilisation Analysis: The MoreSpace Project at TU Vienna.** FBS 19; ISBN 978-3-901608-69-8 (ebook), ASIM/ARGESIM Vienna, 2012; www.tuverlag.at – print-on-demand

Christian Steinbrecher: **Ein Beitrag zur prädiktiven Regelung verbrennungsmotorischer Prozesse.**
FBS 18; ISBN 978-3-901608-68-1 (print), 978-3-901608-72-8 (ebook), ASIM/ARGESIM Vienna, 2010.

Olaf Hagendorf: **Simulation-based Parameter and Structure Optimisation of Discrete Event Systems**
FBS 17; ISBN 978-3-901608-67-4 (print), 978-3-901608-99-5 (ebook), ASIM/ARGESIM Vienna, 2010.

www.tuverlag.at
ISBN 978-3-903024-70-0