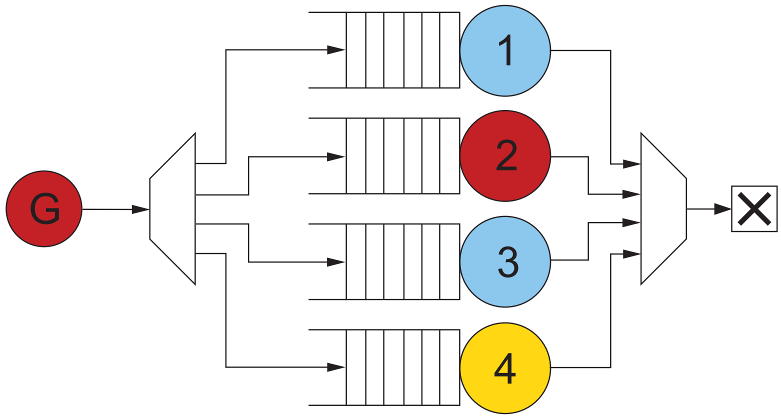
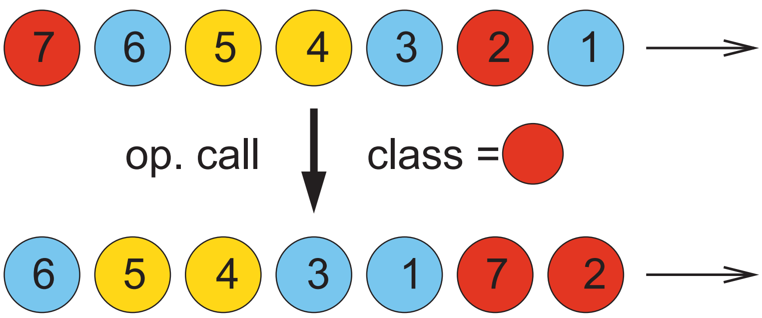
SNE SIMULATION NOTES EUROPE





Journal on Developments and Trends in Modelling and Simulation

EUROSIM Scientific Membership Journal

Vol. 29 No.3, Sept. 2019

ISSN Online 2306-0271
ISSN Print 2305-9974

DOI 10.11128/sne.29.3.1048
ISBN Print 978-3-903024-87-8

TU Verlag an der Technischen Universität Wien

ARGESIM

Letter from the EUROSIM President

Dear colleagues from the modelling and simulation community:

It is a great pleasure for me to write to you as the new president of EUROSIM. This decision of the EUROSIM Board is a great acknowledgement for the Dutch Benelux Simulation Society (DBSS) as one of the oldest simulation societies in the world. DBSS is now 37 years old and it is promoting the active participation of academic scientists, government and industry for providing solutions to challenges that the Dutch society faces actually and in the future.

In recent years, Modelling and Simulation has been getting more and more demand and attention in diverse sectors. This is because it is the only technique that is able to consider the complexity of the systems together with the variability inherent to diverse systems. For this reason, different approaches are applied in diverse systems, discrete, continuous, agent-based, formal methods and hybrid approaches.

We have witnessed the use of M&S in diverse areas like finance, supply chain, information technology, disaster management, security and many more apart from the traditional ones of manufacturing, transport or engineering. Also some firms are using it as value provider for different end-users, new software firms have entered to the market providing novel solutions that combine simulation with other techniques. All these developments increase the demand of experts with M&S knowledge and related skills worldwide and in all the sectors. For the previous reasons, organisations like EUROSIM have a fundamental role in the current and future society to set the framework and arena for knowledge exchange, dissemination of new knowledge and contact between the academic institutions, government entities and industry.

In the next period of 2019-2022 until the congress in Amsterdam, I plan to keep promoting all the activities of the different societies so that the EUROSIM federation acts as an organic organization that evolves during three years towards the most important and relevant activity which will be the EUROSIM congress. The congress will be the maximum event in Europe where simulation scientists, government participants and firms get together in Europe to exchange innovative ideas, applications and discuss the necessities and requirements of innovative applications and areas of study where the use of M&S will make a difference.

In order to achieve this, the executive board will start new activities that encourage a more active participation of the societies for satisfying the necessities that actors of society require from EUROSIM.

Finally on behalf of the new EUROSIM Board and all the EUROSIM societies I would like to thank Prof. Emilio Jiménez and all the members of the EUROSIM Board for their past efforts. I hope that the new board will continue their good work. I would also like to thank everybody who was active in the organization of the very successful EUROSIM Congress in La Rioja.

Dr. Miguel Mujica Mota, m.mujica.mota@hva.nl, EUROSIM President, September 2019

Editorial

Dear Readers *– This issue SNE 29(3) concentrates on two classical SNE topics, on ARGESIM Benchmarks, and on Postconference Publications for EUROSIM societies. Since 1990 SNE is the publication medium for the ARGESIM Benchmarks for Modelling Approaches and Simulation Implementations. We are glad that Peter Junglas and Thorsten Pawletta have mastered the challenge to define the new benchmark 'Non-standard Queuing Policies' and publish the definition in this issue. This new benchmark deals with the handling of the non-standard queuing policies jockeying, reneging, and classing, which are up to now a challenge for discrete simulation systems, on the modelling side, and on the implementation side. We hope to publish many benchmark solutions, benchmark reports, and benchmark studies for this new ARGESIM Benchmark C22. The second benchmark publication in this issue is an educational benchmark study for ARGESIM Benchmark C11: SCARA Robot. This study compares basic MATLAB and EXCEL implementations, carving out essential aspects in modelling and in numerics, and checking features of EXCEL for this type of simulation – a valuable source also for education.*
This issue also starts with postconference publications of last year's ASIM Symposium Simulation Technique. The first selected contributions deal with simulation methods, simulation applications, and education by simulation: system entity structure trees for variability modeling and agent-based simulation of job shop production, simulation-based optimization of generic powder coating lines and a simulation study for the influence of a truck appointment system for the drayage network, and a study on oHMint - an online mathematics course and learning platform for MINT students.

I would like to thank all authors for their contributions to SNE 29(3) - and thanks to the editorial board members, 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, electronic publishing, and much more.

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

Contents SNE 29(3)

Online SNE 29(3), DOI 10.11128/sne.29.3.1048
ARGESIM Publisher, Vienna, www.argesim.org
Print SNE 29(3) ISBN 978-3-903024-87-8,
TU Verlag Vienna, Print-on-Demand, www.tuverlag.at

SNE Contact & Info

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

SNE Editorial Office
Johannes Tanzler (Layout, Organisation),
Irmgard Husinsky (Web, Electronic Publishing),
Felix Breitenecker (Organisation, Author Mentoring)
ARGESIM/Math. Modelling & Simulation Group,
Inst. of Analysis and Scientific Computing, TU Wien
Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria

SNE Simulation Notes Europe

WEB: → *www.sne-journal.org*, DOI prefix 10.11128/sne
Scope: Developments and trends in modelling and simulation in various areas and in application and theory; comparative studies and benchmarks (documentation of ARGESIM Benchmarks on modelling approaches and simulation implementations); modelling and simulation in and for education, simulation-based e-learning; society information and membership information for EUROSIM members (Federation of European Simulation Societies and Groups).
Editor-in-Chief: Felix Breitenecker, TU Wien, Math. Modelling Group *Felix.Breitenecker@tuwien.ac.at*, *eic@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 – only upon invitation, up to 14 p.
- **EBN** Educational Benchmark Note, 2 – 10 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 (increasing) board is taking care on peer reviewing of submission to SNE:

Non-standard Queuing Policies: Definition of ARGESIM Benchmark C22

Peter Junglas[1*], Thorsten Pawletta[2]

[1]Dep. of Engineering "Dr. Jürgen Ulderup", PHWT Vechta/Diepholz, Schlesierstr. 13a, 49356 Diepholz, Germany; *peter@peter-junglas.de

[2]Wismar Univ. of Applied Sciences, Fac. of Engineering, Research Group CEA, PF 1210, 23952 Wismar, Germany

SNE 29(3), 2019, 111-115, DOI: 10.11128/sne.29.bn22.10481
Received: March 10, 2019; Revised: June 26, 2019;
Accepted: July 5, 2019
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna,
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. The ARGESIM benchmark C22 'Non-standard Queuing Policies' studies three non-standard queues that provide different ways to access entities inside a queue like detaching elements or reorder them: The reneging queue, where entities leave a queue after a given waiting time, the jockeying queue, where entities can switch to another shorter queue, and the classing queue, where at certain times entities with a given attribute ("class") are called to the front of the queue. A special focus lies on the management of concurrent events. The benchmark is especially suited for beginners in the field of discrete event modeling.

Introduction

In many applications of discrete event system simulation the modeling of queueing systems is of paramount importance. The basic paradigm describes abstract entities (customers, products, messages) that enter a queue, wait, until the corresponding server is free, and leave the queue, when they are selected for processing according to the queue discipline. Modifying the properties of server processes – such as service time distribution or the possibility of failure – and of the corresponding queues – e.g. their size or discipline – can change the overall system behaviour drastically. Furthermore one is often interested in specific statistical properties of the servers, queues and entities like mean uptime, queue length or average waiting time.

For this reason many simulation environments provide ready-to-use components of standard queues and servers, either as different blocks like in SimEvents [1] or combined in one `Process` module as in Arena [2]. Using parameters one can easily define the size of a queue, choose among a set of predefined disciplines (e.g. FIFO, LIFO, priority) or change the service time distribution.

But there are a lot of important examples, where the behaviour of the queue selection process or of the entities in the queue is more complex [3]. In this benchmark we will concentrate on the following three scenarios:

- *Jockeying:* the last entity in one of a set of FIFO queues can switch to another shorter queue.
- *Reneging:* entities wait in the queue only for a fixed maximal time and leave the queue and the system, if they are not served before.
- *Classing:* entities have a class attribute, similar to a priority, and advance to the front of the queue, when an external operator calls for their class number.

In a standard FIFO queue, an entity can only leave when it reaches the front of the queue. In contrast the three examples introduce additional ways to access entities inside a queue: detach the last element (jockeying), detach any element (reneging) or reorder all elements (classing). Frequently the standard queue components defined in simulation environments do not offer such access. This can make the implementation of a non-standard queue rather complicated, introducing a lot of additional internal events [4].

All three examples are variations of a simple basic queueing system containing four standard FIFO queues and servers. Its implementation should be straightforward, but it can be useful nevertheless, in order to scrutinize the exact system behaviour in the case of concurrent events. In the context of mainly stochastical processes this seems to be an unlikely case, but it is important e. g. to control the exact order of event cascades [5].

Therefore the models in this benchmark have to be implemented in two versions: A smaller deterministic model allows for exact comparisons and outputs plots of its dynamic behaviour, while a larger stochastic model produces statistical data of some relevant system variables.

1 Basic Queuing System

The simple queueing system shown in Figure 1 forms the basis of all examples. Its main purpose is to define all the details that are identical for the following special cases, but it also allows for some interesting variations that will be studied in the benchmark.

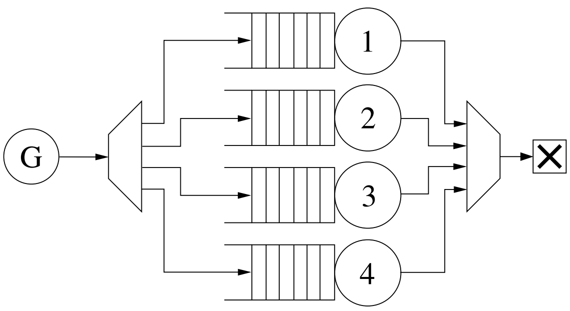


Figure 1: Basic queueing system with four queues.

The system contains a generator that creates a given number n_E of entities with fixed or stochastic interarrival times t_A, the first one starting always at $t = 1$. The entities have id attributes $1 \dots n_E$ that are given in order of creation. They enter the system of $n_Q = 4$ numbered queues and servers, choosing the shortest line, including the server allocation. In case of several queues with minimal length the one with the smallest number is chosen. The queues have a FIFO ("First In First Out") discipline and a potentially infinite capacity. Each server has a capacity of one and a fixed or stochastic service time t_S.

After being served the entities leave the system, e. g. they are terminated. The simulation stops, when all n_E entities have been terminated.

The *deterministic version* has $n_E = 100$ entities, constant interarrival time $t_A = 1$ and constant service time $t_S = 4.5$. Its simulation should produce two plots, one showing $ids(t)$, i.e. the ids of the last 20 outgoing entities over their termination time, preferably as stem or bar plot, the other the total queue length $l_{qt}(t)$ (i.e. the sum of the four queue lengths, without the server allocation) over the complete simulation time.

The *stochastic version* uses $n_E = 500$ entities, the interarrival times are exponentially distributed with a mean value of $t_A = 1$. The service times are computed using a symmetric triangular distribution with the most likely value $t_S = 4.5$ and a half-width $\Delta t_S = 2$, i. e. the possible values range from $t_{S,min} = 2.5$ to $t_{S,max} = 6.5$. The simulation should output the maximal and the average value of the total queue length $l_{qt}(t)$ (again defined as the sum of the four queue lengths, without the server allocation), where the average is defined as time average over the complete simulation time. Additional results are the maximum and average of the queue waiting times $t_{q,i}$ of entity i, where the service time is not included and the average is taken over all entities.

These two base models are pretty much standard, their implementation should present no difficulties. The two versions should be basically identical, especially no optimisations are allowed that depend on the special parameters of the deterministic input or server process.

The above description is not complete, insofar as the concrete order of concurrent events is not specified. This is an important issue at least for the deterministic version, where the results actually depend on such fine details. Therefore the concurrency order is generally specified in the following way:

1. an entity leaves a server,
2. a queued entity enters a server,
3. a new entity enters the system and chooses a queue.

For the variants described in the following sections additional event types may occur and the concrete order will be fixed accordingly. To study the methods, how the order of concurrent events can be fixed in a simulation environment, this benchmark includes another, optional model, namely a variant of the deterministic basic model with *another concurrency order*:

1. a new entity enters the system and chooses a queue,
2. an entity leaves a server,
3. a queued entity enters a server.

Another interesting problem is, how a simulation program deals with *large systems*. Especially for standard graphical environments the creation of a lot of queues and servers with copy and paste is a nuisance, there should be better ways to cope with the size. Therefore the benchmark includes another optional model, a variant of the stochastic version with 40 queues and servers. All parameters are as before except for the following:

$$n_Q = 40, \quad t_A = 0.1, \quad n_E = 5000.$$

2 Jockeying Queues

In a system with several queues *jockeying* means the process that an entity moves from one queue to another, usually shorter queue [3]. In the context of this benchmark it is specified in detail as follows:

- Jockeying happens immediately, when a queue (incl. server) is at least shorter by 2 than another one.
- In this case the last entity of the longer (*source*) queue leaves its queue and becomes the last entity of the shorter (*destination*) queue.
- If there are several destination queues, the one with the smallest queue number is chosen.
- In case of several possible source queues, the one is chosen that is nearest to the destination queue, i.e. where the absolute value of the difference of their queue numbers is minimal. If there are two such source queues (one on each side of the destination queue), the one with the smallest number is selected.

To complete the specification the order of concurrent events is given as:

1. an entity leaves a server,
2. a queued entity enters a server,
3. a jockeying entity changes its queue,
4. a new entity enters the system and chooses a queue.

Both variants of the basic model have to be augmented with the described jockeying behaviour. All parameters remain the same, as well as the standard output graphs and statistical values. For a jockeying entity the queue waiting time is defined as the sum of its waiting times in all queues it has visited (possibly many). Additionally the stochastic model should output the total number of jockeying events, while the output of the deterministic version should include a table showing the time of each jockeying event, the id of the jockeying entity and the numbers of the source and the destination queue. For conciseness, only the first five and the last five rows of the table have to be included in a benchmark report.

3 Reneging Queues

In this model entities that have entered a queue can leave it, before they are being served. This behaviour is called *reneging* [3]. There are many possible strategies, when entities renege, but in the benchmark example it is simply done, when the maximal waiting time $t_R = 9$ is reached. The order of concurrent events is as in the standard case, with the additional requirement that entering a server takes precedence over reneging, i. e. the order is:

1. an entity leaves a server,
2. a queued entity enters a server,
3. a queued entity reneges,
4. a new entity enters the system and chooses a queue.

Again both versions of the basic queueing system have to be extended to include the reneging of entities, using identical parameters and output graphs resp. values. For the computation of the average queue waiting time the time of the reneging entities – being t_R of course – is included. In addition the deterministic model should output a table of time and id of all reneging entities, while the stochastic model simply shows their total number.

4 Classing Queues

The last example system, called *classing queues*, is the most complex. It is inspired by a typical situation during the boarding of a plane: An operator calls "all passengers with seat numbers 15 – 30" to the front of the queue.

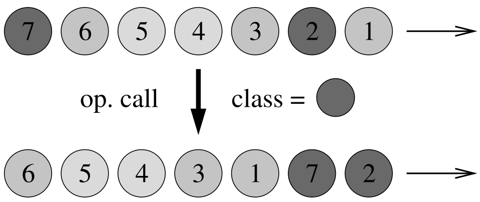


Figure 2: Result of an operator call (class ≙ color).

To mimic this each entity i is supplied with a fixed integer class c_i, where $1 \le c_i \le n_C$, with the number of classes given as $n_C = 5$. At certain times an operator calls for a class number, whereupon all entities with this class procede to the front of their queues. The relative order of the entities within this class remains intact, as does the order of the other entities among themselves (cf. Figure 2). Such a behaviour is similar to the standard priority queue, with the essential difference, that the meaning of "high priority" changes at runtime.

As before the classing behaviour has to be included in both variants of the basic queueing system. The way entities are assigned their class is different for the variant models: In the deterministic case the classes c_i are dealt in a round robin way in ascending order, starting with 1. In the stochastic version they are chosen randomly with equal probability $1/n_c$ for each class. An entity can only proceed to the server, if its class is the currently called class.

Unlike the boarding example, which ends after the boarding process is completed, the here defined "classing queue" should possibly run forever. Therefore the exact behaviour of the operator is defined as follows:

- Initially it waits for a fixed time $t_C = 10$, during which all incoming entities remain in the queue.
- After that it calls the classes in descending round-robin order, starting with the highest one ($n_C = 5$). This defines the current class identically for all queues, i.e. there is only one operator for the whole system.
- After a call it waits, until all entities of the current class have been served, before it calls the next class.
- If there are no entities in the system, the operator is stalled, until a new entity arrives.

The call of the operator has lowest priority among concurrent events, apart from that the order is as in the standard case:

1. An entity leaves a server,
2. a queued entity enters a server,
3. a new entity enters the system and chooses a queue,
4. the operator calls a new class and the queue is reordered accordingly.

The standard output graphs resp. statistical values have to be supplied, together with a table of the average and maximal values of the queue waiting times $t_{q,i}$ for each class, in both variants.

5 Specification of all Benchmark Tasks

All benchmark models have been described above with their exact parameters, together with the requested outputs of corresponding simulation runs. For easier reference, this section summarizes all items that a benchmark report should contain.

Basic Queuing System

1.1 A short description of the relevant parts of the basic model, using plots of the component structure, code snippets or whatever may be appropriate to understand the basic idea of the implementation.

1.2 plots of $ids(t)$ and $l_{qt}(t)$ (deterministic model),

1.3 results for max. and avg. of $l_{qt}(t)$ and $t_{q,i}$ (stochastic model),

1.4 (optional) a comparison of the implementations of the two concurrency variants together with a plot of $ids(t)$ for the variant model,

1.5 (optional) a comparison of the implementations of the standard and large stochastic models together with max. and avg. of $l_{qt}(t)$ and $t_{q,i}$ for the large model.

Jockeying Queues

2.1 A short description of the implementation of the jockeying queues,

2.2 plots of $ids(t)$ and $l_{qt}(t)$ and a table displaying the first five and the last five jockey events (deterministic model),

2.3 results for max. and avg. of $l_{qt}(t)$ and $t_{q,i}$ and the number of jockey events (stochastic model).

Reneging Queues

3.1 A short description of the implementation of the reneging queues,

3.2 plots of $ids(t)$ and $l_{qt}(t)$ and a table displaying the reneging events (deterministic model),

3.3 results for max. and avg. of $l_{qt}(t)$ and $t_{q,i}$ and the number of reneging entities (stochastic model).

Classing Queues

4.1 A short description of the implementation of the classing queues,

4.2 plots of $ids(t)$ and $l_{qt}(t)$ and a table displaying class statistics (deterministic model),

4.3 results for max. and avg. of $l_{qt}(t)$ and $t_{q,i}$ and a table displaying class statistics (stochastic model).

Solutions should be accompanied by the complete source code of all models to make them accessible on the ARGESIM Benchmark server.

6 Conclusion

Depending on the simulation environment used, some of the tasks can be very easy or may require tricky modeling and implementation ideas. Probably a special difficulty will be to guarantee the specified order of concurrent events. The production of plots and statistical results tests the corresponding capabilities of the simulation environment, but can of course be done using data export and external programs.

Since all system examples are rather small and don't need a special mathematical or modeling background, the benchmark is suited for beginners in the field of modeling and simulation.

Acknowledgement

The first author (P. J.) likes to thank the team of the CEA research group in Wismar for the warm hospitality extended to him and many helpful discussions.

References

[1] Li W, Mani R, Mosterman P. Extensible discrete-event simulation framework in SimEvents. *Proc. 2016 Winter Simulation Conference*; 2016 Dec; Arlington. New Jersey: IEEE. 943-954.

[2] Kelton W, Sadowski R, Zupick N. *Simulation with Arena*. 6th ed. New York: McGraw-Hill; 2015. 656 p.

[3] Hassin R, Haviv M. *To queue or not to queue: Equilibrium behavior in queueing systems*. Boston: Kluwer Academic Publishers; 2003. 191 p.

[4] Austermann L, Junglas P, Schmidt J, Tiekmann C. Conceptional problems of transaction-based modeling and its implementation in SimEvents 4.4. *Simulation Notes Europe SNE*. 2017; 27(3): 137–142. doi: 10.11128/sne.27.tn.10383

[5] Junglas P. Pitfalls using discrete event blocks in Simulink and Modelica. In: *Proc. of ASIM-Workshop STS/GMMS*; 2016 Mar; Lippstadt. 90-97. ISBN 978-3-901608-48-3.

How to Define SES Trees for Variability Modeling

Christina Deatcu[1*], Hendrik Folkerts[1], Thorsten Pawletta[1], Umut Durak[2]

[1]Hochschule Wismar - University of Applied Sciences, Research Group CEA, Philipp-Müller-Straße 14, 23966 Wismar, Germany; *christina.deatcu@hs-wismar.de

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

SNE 29(3), 2019, 117 – 126, DOI: 10.11128/sne.29.tn.10482
Received: May 10, 2019 (Selected ASIM SST Hamburg 2018 Postconf. Publ.), Accepted: June 10, 2019
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. The System Entity Structure (SES) is a high level approach for variability modeling, particularly in simulation engineering, which is under continuous development. In this context, an enhanced framework is introduced that supports dynamic variability evolution using the SES approach. However, the main focus is to start a discussion about a set of design patterns, which were developed to analyze the tree design and computing aspects of System Entity Structures. As development of our MATLAB-based SES toolbox for construction and pruning of SES trees proceeded, the necessity to have some generalized examples for testing and verification came more and more into awareness. We propose a set of design patterns that, if completely representable and computable by a certain tool, support all aspects of SES theory. In addition, the patterns give users substantial support for developing SES models for other applications.

Introduction

This paper is a modified version of [1]. It details the specification of basic design patterns and introduces more advanced combined patterns.

Generally, variability modeling can be seen as an approach to describe more than one system configuration. According to Capilla, Bosch, and Kang [2], a software variability model has to describe the commonality and variability of a system at all stages of the software lifecycle. In software engineering, variability modeling is often closely associated with product lines. For software product lines the variability is described explicitly. Variation points are defined where different solutions can be derived.

Such variability mechanisms can be specified at different levels of abstraction, ranging from requirements specification to source code implementation. A popular high level approach is feature modeling by means of feature models, which were introduced as part of Feature-Oriented Domain Analysis by Kang et al. [3] and subsequently extended and used in various ways. An important further development of variability modeling has been the notion of variability in time, known as binding time in product line engineering [4]. That means that variability can be realized from design time to runtime.

In simulation engineering, the problem of variability modeling is well known from the eighties. One of the first high level approaches for variability modeling in the design phase was introduced with the System Entity Structure (SES) by Zeigler [5]. The objective was to describe a set of system configurations for a family of systems. An SES is represented by a tree structure, which describes a set of modular, hierarchical system structures, defines references to basic models in a model base (MB) and specifies various parameter settings for the referenced basic models. In addition, the approach defines several abstract transformation methods for deriving a particular system configuration and for generating an executable simulation model [6]. The entire approach was continuously further developed by Zeigler and many other researchers, such as in [7], [8], and [9].

As another approach for describing system structures and their configurations, various XML based composition schemes, which distinguish between interfaces and the concrete implementations of the models were proposed, such as by Röhl and Uhrmacher [10] or by Wang and Wainer [11]. Although that research does not explicitly address model families as the SES/MB approach does, some of the basic ideas are similar. Moreover, the ideas in [10, 11] are important for the design of reusable model components and their organization in an MB. In this paper, the reusability of components in an MB is not discussed.

It focuses on modeling system variability using the SES and the necessary software framework.

The problem of variability at runtime is known as variable or dynamic structure system modeling and simulation. Analogous to the approaches in software engineering, such a dynamic variability can be described on the level of a specific model, such as introduced by Barros in [12], or separated using a higher level model abstraction in conjunction with an appropriate software framework. Regarding this, a first theoretical approach for dynamic variability modeling using the SES/MB method was published in 1990 [13].

Based on this early idea, a prototype of a full SES/MB based modeling and simulation infrastructure has been developed and implemented within MATLAB/Simulink by the authors [14, 15], and a Python implementation is in progress. In addition, the SES theory has been enhanced. In this context, the necessity to have some generalized SES patterns came more and more into awareness. On the one hand patterns are helpful during software development and on the other hand the patterns are expected to give users substantial support for developing SES models for their applications.

After a short overview to the enhanced SES ontology and the enhanced SES/MB based infrastructure, basic design patterns for variability modeling are introduced. The description is related to the modeling capabilities provided by feature models. Then, some combined patterns are discussed exemplary to give an impression for advanced variability modeling possibilities using the SES and its extensions. Finally, the main results are summarized and an outlook to future work is given.

1 Background

According to Zeigler and Hammonds, the SES is an ontology, a language with syntax and semantics to represent declarative knowledge [8]. It is particularly suitable for describing system configurations for different application domains. An SES is represented by a directed tree structure. Objects are represented by nodes which are connected by edges. There are four node types with different properties describing the objects and their relations. Furthermore, there are axioms for defining the SES correctly. Since an SES describes a number of system configurations, the SES tree needs to be *pruned* to get one particular configuration, which is called Pruned Entity Structure (PES).

The classic SES theory was extended by several researchers over the last decades. In [14] and [16] the SES theory was extended with a procedural knowledge representation. Some of these extensions are used in this paper. A comprehensive example on how the pruning patterns proposed in this paper can be used, is demonstrated in [17].

1.1 Node Types

Among the four node types, there are two groups, the entity nodes and the descriptive nodes. Entity nodes describe objects of the real or the imaginary world. The root and the leaves of an SES tree are always entity nodes. Relations between the entity nodes are specified by descriptive nodes.

Descriptive nodes are the genus for aspect nodes, specialization nodes and multi-aspect nodes. Aspect nodes (name suffix *DEC*) describe how entity nodes can be decomposed in partial entities whereas the taxonomy of an entity is described by specialization nodes (name suffix *SPEC*). Multi-aspect nodes (name suffix *MASP*) are a special case of an aspect node with all children being of the same kind.

Each node or edge can have attached variables, also called attributes. For entity nodes, the variables represent properties of the respective object whereas the variables at descriptive nodes specify relations between their parent node and children nodes or decisions for the pruning process. With the extended procedural knowledge representation, values of attached variables can be assigned dynamically.

1.2 Axioms

The semantics of the SES are defined by axioms. The types of the nodes have to follow the axiom *alternating mode*. Every entity node has to be followed by a descriptive node, and vice versa. A *strict hierarchy* is needed. In every path of the tree, a name of a node may occur only once. If nodes in different paths have the same name, they need to have the same variables and isomorphic partial trees. This is called *uniformity*. Nodes on the same level of hierarchy and having the same father, called sibling nodes, have to be *valid brothers*, meaning that sibling nodes must not have the same name. The axiom of *attached variables* implies that a node must not have variables of the same name. The axiom of *inheritance* implies, that during pruning, the parent and the child of a specialization combine their attributes.

If parent and child have the same attributes, the parent's attributes are overwritten with the child's attributes and their values.

1.3 Extended SES/MB Infrastructure

The SES describing a set of system designs has been associated with the idea of model generation of modular, hierarchical systems from the very beginning [6] which led to the SES/MB approach. Each system design is defined by its system structure and parameter configuration in the SES. The core assets of all system variants are specified as a set of configurable basic models, which are organized in a Model Base (MB). The classic SES/MB framework defines a set of transformation methods for generating executable simulation models, but automated model generation is not provided. To allow automated generation and execution of models, the SES/MB approach has been extended ([14], [15], [16]). These extensions make the SES/MB approach more pragmatic for implementation and to be used in a simulation infrastructure.

Figure 1 depicts the extended SES/MB infrastructure consisting of the SES/MB framework, an *Execution Unit*, and an *Experiment Control*. Although the SES/MB approach and its extensions are usually considered in connection with the generation of simulation models, they are generally applicable to modular-hierarchical structured software systems.

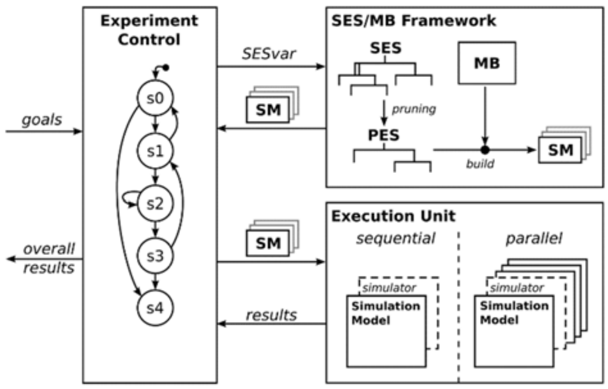


Figure 1: Extended SES/MB-based infrastructure.

Operations. On the SES, a *merge operation* is defined allowing two or more SES to be combined. This allows the quick reuse of a once defined SES. The essential operation on the SES is the *pruning method*. To extract one particular system structure and configuration, the SES needs to be trimmed to a PES. During the pruning process, decisions have to be taken at descriptive nodes.

Therefore, rules need to be defined at aspect, multi-aspect and specialization nodes. The *specialization rule* (specrule) associated with a specialization node determines which child entity shall be selected. *Aspect rules* (aspectrule) associated with aspect or multi-aspect nodes on the same hierarchy level determine which of the siblings is to be chosen. Furthermore, cross-tree relations can be expressed by *selection constraints*. Selection constraints can be used to select a certain entity based on decisions taken anywhere else in the SES tree. Next to the pruning method, another transformation method is the *build method*. With the help of the build method, an executable model can be built from a PES and basic models organized in an MB. The basic models are specific for a certain simulation software. Therefore, the build method needs to match to the simulator used.

Execution Unit and Experiment Control. For automated and reactive processing of SES models, an execution unit and an overall experiment control unit are added to the framework, as depicted in Figure 1. For automatic generation of different PES, leading to different simulation models, an interface to the SES is needed. This interface can be established by global variables of the SES, called *SES Variables* (SESvar), which can affect the decisions taken in descriptive nodes during pruning. Thus, a particular system configuration derived from an SES depends on the current settings of the SES variables. The value range of SES variables can be limited by defining *semantic conditions*, which are checked before pruning to exclude certain system configurations. By assigning values to the SES variables, the experiment control determines the order and system configurations of executable simulation models (SM) to generate from the SES with the pruning and build operations. Thereby different variants of the executable simulation models are generated. The experiment control then transmits the SM to the execution unit. The execution unit links the generated simulation model to the simulator, executes a simulation run and, finally, sends the results back to the experiment control. The results, in turn, can influence the decision of the experiment control on how to assign the SES variables next.

Special Attributes. Combining basic models from the MB leads to the creation of *coupled models*. In order to describe the structure of the executable model, some nodes need to define *couplings*. Couplings are properties of descriptive nodes of the type aspect and multi-aspect and consist of pairs of entity names and port names. Figure 2 gives an impression of what a definition of couplings may look like. Furthermore, for a multi-aspect node, a special variable, *numRep*, has to be defined representing the number of children to generate when pruning this node. To specify the basic model from the MB an entity node refers to, the *mb-attribute* is introduced. This special attribute is permitted just for leaf nodes. Finally, for some cases, it is necessary to define priorities for supporting decisions among descriptive nodes on the same level of hierarchy in the *priority attribute*. All values of attributes can be defined by constants or set via SES variables or SES Functions.

SES Functions. The concept of *SES Functions* (SESfcn) has been introduced to specify complex variability within node attributes with minimal effort and to keep a lean SES tree. Typical examples include the definition of varying coupling relations, varying port numbers of systems or the definition of variable parameter configurations in attributes. During pruning, SES functions are evaluated, often with SES variables as input parameters. For effective coding of SES functions, the implicit attributes *parent* and *children* are introduced for each SES node. They encode the parent and children node names, respectively.

1.4 Software Tools for the Extended SES/MB Infrastructure

In the Research Group CEA, a prototype tool for the SES/MB infrastructure was developed, *The SES Toolbox for Matlab/Simulink* [15]. Currently, SES trees can be defined via a graphical user interface and a concrete variant can be extracted by pruning. The toolbox supports the modeler with plausibility test during SES construction, graphical representation of the SES, automatic generation of HTML documentation, and other features. The pruning process can be started from the graphical user interface and, in addition, is implemented to function automatically. Automatic pruning is necessary when using an SES constructed with the toolbox together with the experiment control. Furthermore, there is a prototype Matlab function implementing a build method for the simulation software Simulink, including SimEvents and Simscape, the MatlabDEVS toolbox [18], and for Modelica models. The SES is linked to the appropriate MB with the special mb-attribute of the leaf entity nodes.

Another software tool based on Python3/PyQt5 is under development. The aim is to be more independent from a computing environment and to support a greater number of simulators for building executable simulation models.

2 Basic Design Patterns

A few elementary design patterns that are required for design, implementation, and test of a pruning algorithm for an SES are presented in the following subsections. The proposed patterns for modeling of system structures are necessary, particularly if one aims to use the SES tool for automated model generation in the context of the extended SES/MB approach. In analogy to the semantics of feature models ([2], [3]) and mathematical logical expressions, we try to classify the first patterns according to their purpose.

In the context of feature modeling, four kinds of features are used: (i) mandatory features (logical AND), (ii) alternative features (logical XOR), (iii) optional features and, (iv) OR-features (logical OR). At first sight, it seems to be obvious and simple to decide which SES constructs can be used to define tree sections satisfying the feature categories above, but there is often more than one way to express the logical relations among tree sections, components or entities. However, the simplest patterns consist of just one descriptive node with its parent and children.

Besides patterns fitting into the classification according to feature modeling, we identified some patterns useful to illustrate and test SES tools dealing with issues like inheritance, the special attributes, such as couplings and priorities, and evaluation order.

2.1 Mandatory Tree Sections

Mandatory are those sections which need to be existent in each system belonging to a family of systems. For an SES describing a family of systems, this means that all possible PES representing a certain system variant will also include those parts. The corresponding logical expression is the AND and we can call the linkage a '*has-a*' relationship. Design patterns of this type are the aspect node itself, a multi-aspect node or specialization siblings.

Design Pattern #1 - Aspect Node. Figure 2 depicts the simplest case of a design pattern for mandatory sections, the aspect node itself.

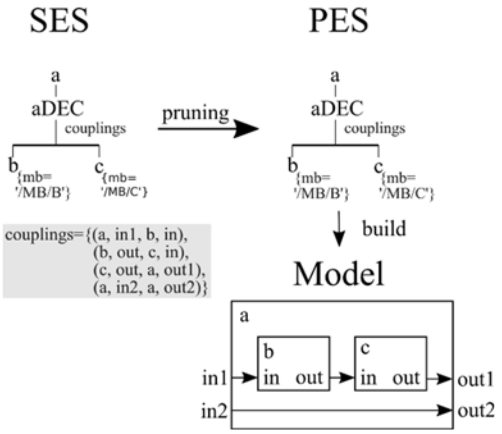


Figure 2: First design pattern for mandatory sections.

Each coupled system *a* consists of an entity *b* and an entity *c*. For model generation, aspect nodes need to define the special attribute for couplings, while the leaf nodes have the mb-attribute attached. Note that the derived PES is identical to the SES. The resulting abstract, simulator independent model on the right side is given to illustrate what kind of model structure would result from the PES. The following examples will not include detailed coupling definitions and models since couplings can be formulated analogously and concrete models depend on the chosen simulation environment.

Design Pattern #2 – Multi-Aspect Node. In Figure 3, a similar case to design pattern #1 is given. With this pattern we introduce the usage of SES variables, SES functions and the possibility to define multisets for setting values of attributes. Since the system *as* consists of a certain number of children of type *b*, multi-aspect nodes need to define two special attributes, the attributes numRep and couplings. When pruning a multi-aspect node, the numRep attribute is evaluated and an aspect node with children of the same type is created. In this example, the numRep attribute is defined by the current value of the SESvar VAR which is restricted to two or three by the semantic condition, but the value could also be hardcoded. Based on the number of children the couplings may vary, too. For an effective coding and to keep a lean SES tree, couplings are set via an SESfcn here. Based on the value of the SESvar VAR either cpl1 or cpl2 is chosen. The structure of cpl1 and cpl2 needs to be defined as it is presented for static couplings in Figure 2.

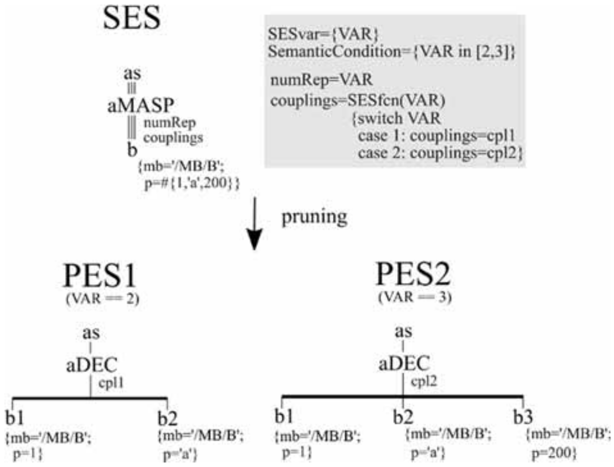


Figure 3: Design pattern for mandatory sections with a multi-aspect node.

The attribute p is set by dint of a multi-set variable. Although after pruning all children of *as* are of the same type *b* and are referencing the same basic model B in the MB, their parametrization can be different. Children's names are generated by appending a number to the entity name *b* to ensure that the resulting siblings fulfill the axiom of valid brothers.

Design Pattern #3 - Specialization Siblings. If two or more specialization nodes are on the same hierarchy level, they will all be evaluated. For the pattern depicted in Figure 4, this means that *a* will specialize into one of *b* or *c* AND into one of *d* or *e*. Which child of the specialization is taken depends on the values of the SES variables and the specialization rules.

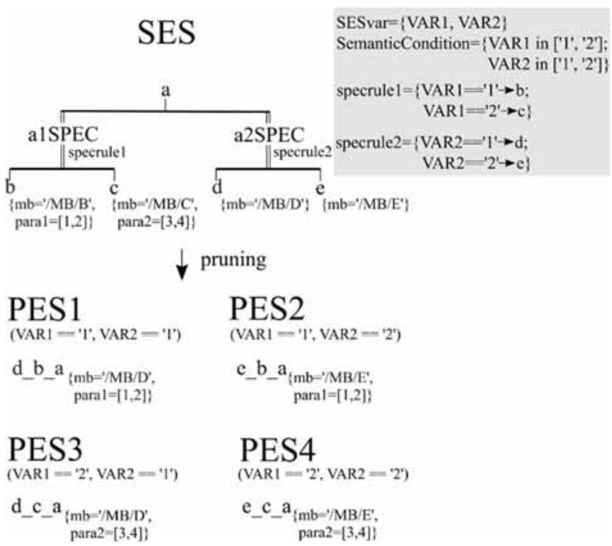


Figure 4: Design pattern for specialization siblings.

If, for example, VAR1 is set to 1 and VAR2 is set to 2, at specialization *a1SPEC* the left child *b* is selected and, at specialization *a2SPEC*, the right child *e* is selected. During pruning, it depends on the pruning algorithm which of the two specializations is evaluated first. For this example, we assume that the left specialization node *a1SPEC* is evaluated first. The evaluation order influences what is the resulting value of the mb-attribute since, according to the inheritance axiom, attributes with the same name are overwritten in the parent node.

With this example, we also demonstrate the use of SES variables for defining specialization rules.

2.2 Alternative Tree Sections

Alternative are those sections from which exactly one part needs to be existent in each system belonging to a family of systems. A system described by a PES is of the type which is selected in a specialization or consists of the children of aspects chosen at aspect siblings. The corresponding logical expression is the XOR and we can call the linkage an '*is-a*' relationship. Design patterns of this type are the specialization node itself, aspect siblings, multi-aspect siblings or siblings containing an aspect node and a multi-aspect node.

Design Pattern #4 - Specialization Node. The simplest pattern to specify alternatives is the specialization node shown in Figure 5. Based on the specialization rule, exactly one child needs to be selected to construct a valid variant. The specialization rule evaluates the SESvar VAR to allow a selection. After pruning, the resulting system can either be of type *b* OR of type *c*. Valid names are constructed by preceding the fathers name *a* with either *b* or *c* and attributes are inherited as defined by the SES axioms.

Design Pattern #5 - Aspect Siblings. If two or more aspect nodes are on the same hierarchy level, exactly one of them has to be selected by evaluating the aspect rules of the aspect brothers.

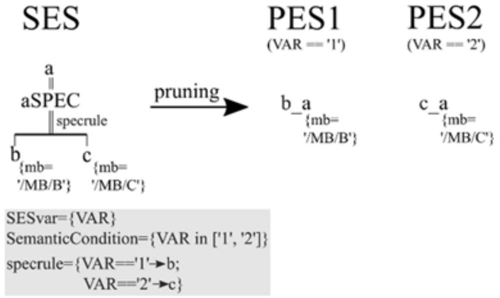


Figure 5: A simple specialization node.

This is presented in Figure 6. The system *a* consists either of *b* and *c* or of *d* and *e*. The aspect rules one and two define which branch is selected during pruning.

Design Pattern #6 – Multi-Aspect Siblings. In the pattern shown in Figure 7, two multi-aspect nodes are on the same layer. In the first pruning step, the multi-aspect nodes are resolved leading to two aspect nodes. After this step, the resulting intermediate PES can be finally resolved using pattern #5 for aspect siblings, as previously described. The system *a* consists either of *b1*, *b2*, and *b3* or of *c1* and *c2* depending on which child is selected based on the aspect rules.

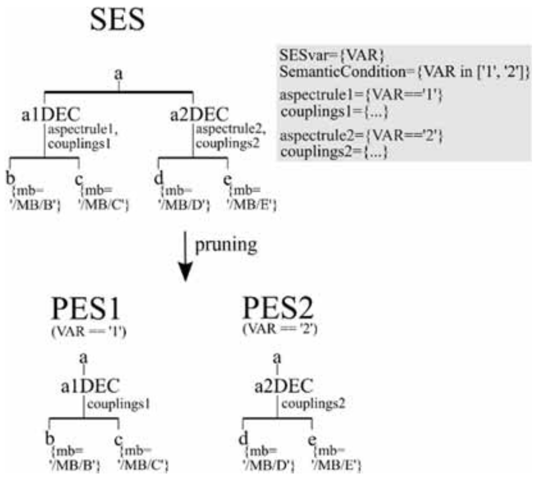


Figure 6: Aspect siblings result in an alternative selection.

Design Pattern #7 - Aspect and Multi-Aspect Siblings. If there are more than one aspect nodes and multi-aspect nodes on the same hierarchy level, the behavior is like aspect siblings after the multi-aspect node is resolved (see Figures 6 and 7).

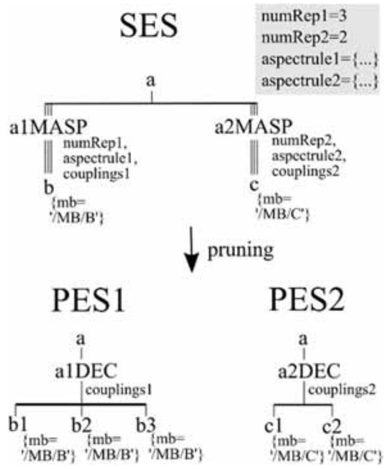


Figure 7: Multi-aspect siblings behave like aspect siblings.

2.3 Optional Tree Sections - Design Pattern #8

Optional sections can be contained in the resulting system structure, but they do not have to be. This can be done using an extension of the SES, the NONE element. A NONE element for a leaf entity node means that, if the NONE branch is selected, the entity is not included at all. In the pattern shown in Figure 8, the specialization has a child which is a NONE element. Hence, this SES can evaluate to NONE during pruning based on the specialization rule. The system *a* can either be of type *b* or not existent at all.

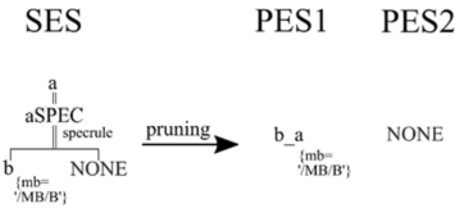


Figure 8: Optional sections expressed by specialization nodes with the NONE element.

2.4 OR Tree Sections - Design Pattern #9

Logical OR means that one or more entities or tree sections need to be included to get a valid variant. This can be expressed by an aspect node whose children are followed by specializations. Each specialization contains a NONE element as one child. The corresponding design pattern is shown in Figure 9. At least one specialization has to evaluate to a node not being NONE. By defining the specialization rules reasonably, the user has to ensure, that this is guaranteed. This pattern is composed by pattern #1 describing mandatory tree sections in combination with pattern #8 for optional elements. After pruning, the system *a* consists of *b* and *c*. System *b*, in turn, is of type *bs* or not existent while *c* is of type *cs* or not existent.

Couplings have to be adjusted when the tree changes by evaluating nodes. Since the couplings are defined at the aspect node *aDEC*, it is obvious, that there is a need for the possibility to define variable couplings. Variable couplings can be defined via SES Functions as introduced with design pattern #2.

3 Combined Design Patterns

In the previous section, the elementary design patterns were discussed in particular. During tool development, the necessity of testing combinations of basic patterns (combined patterns) was recognized.

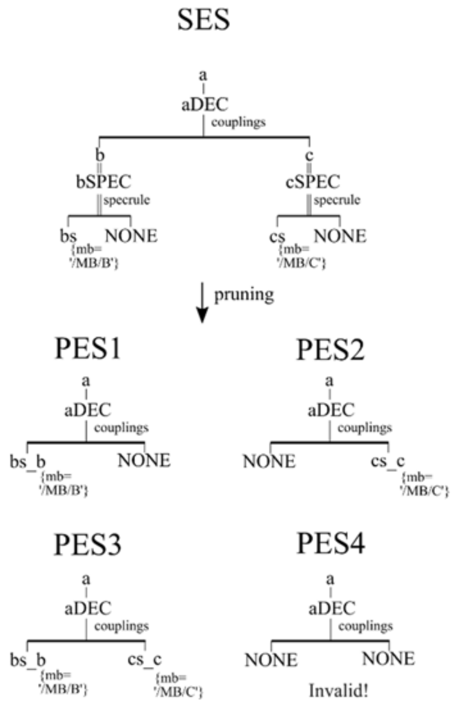


Figure 9: OR expressed by aspect nodes followed by specializations with NONE elements.

Issues addressed with these patterns are inheritance, evaluation order and priorities. In the next sections, a few combined patterns are introduced.

3.1 Design Pattern #10 - Two Specialization Nodes in One Path

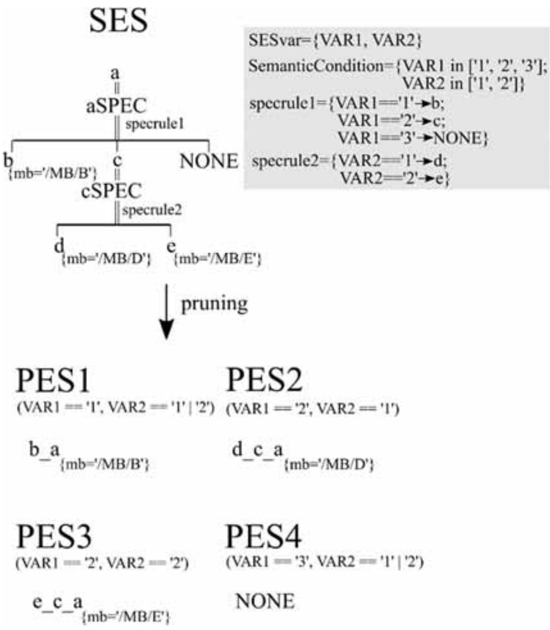


Figure 10: Inheritance of attributes at two specialization nodes in one path.

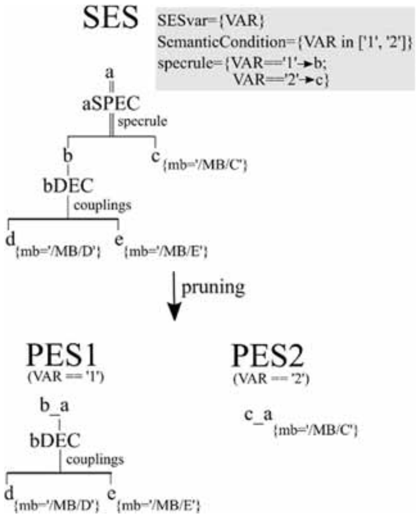


Figure 11: A specialization with a succeeding aspect.

Specialization nodes inherit the attributes of the selected child and append them to the father's attributes, as previously described. In Figure 10, there are two specialization nodes in one path. Additionally, the NONE element is used. Thus, this can be classified as an optional feature and shall clarify the axiom for attribute inheritance. During pruning, firstly *aSPEC* is evaluated. In case the child *c* is selected, another decision for child *d* or child *e* is taken. In that case, the attributes of the child node of *cSPEC* are inherited to the first node.

3.2 Design Pattern #11 – Specialization with Succeeding Aspect

Figure 11 depicts a specialization node with a succeeding aspect node. This pattern is a combination of a specialization node followed by a single aspect node. Therefore, it is a combination of alternative and mandatory tree sections. The system *a* can be of type *b* or *c*, the *b* can be decomposed in *d* and *e*.

3.3 Design Pattern #12 - Specialization and Aspect Siblings

When aspect nodes and specialization nodes are brothers, the specialization node has to be resolved first during pruning. If, additionally, an aspect node is below the specialization node, during pruning two aspect nodes will become siblings. Since, in this case, the occurrence of aspect siblings is not known until the first pruning step, aspect rules could not be formulated beforehand. In order to tackle this, the priority attribute for aspect and multiaspect nodes was introduced. Throughout the whole SES, these nodes get a unique number. If, during pruning, aspect nodes become brothers, a decision about which to choose can be made.

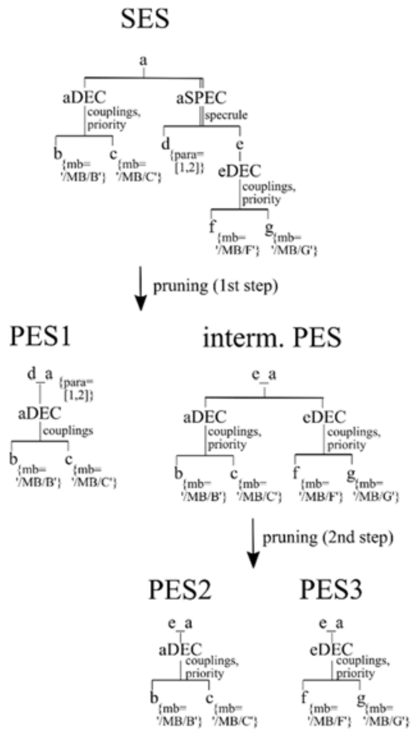


Figure 12: Aspects can become siblings by resolving a specialization.

In Figure 12, an example is given. The system *a* consists of *b* and *c*, if it is of type *d*. If it is of type *e*, it can consist of *b* and *c* or of *f* and *g*. In case the system is of type *e*, a decision as to which decomposition to take is made by interpreting the priority attribute.

3.4 Design Pattern # 13 - Several Multi-Aspects in a Path

If a multi-aspect node is followed by a second or even more multi-aspects, complexity of resulting structures grows considerably. During pruning compliance with the SES axioms has to be ensured and therefor some renaming operations are necessary.

Figure 13 depicts two successive multi-aspects where the numRep variable of *bMASP* defines a multiset. Renaming of *b* to *b1* and *b2,* as necessary for children of multi-aspects and explained in pattern #2, results in renaming of *bMASP* to *b1MASP* and *b2MASP*. Generally, one can say that renaming an entity node always calls for renaming the following descriptive node, too. The first pruning step resolves the multi-aspect *aMASP* to an aspect *aDEC* with two child nodes of type multi-aspect. In a second step the child nodes are resolved to aspect nodes, too. Since the numRep variable of *b1MASP* and *b2MASP* was set via the multiset, variable couplings are necessary.

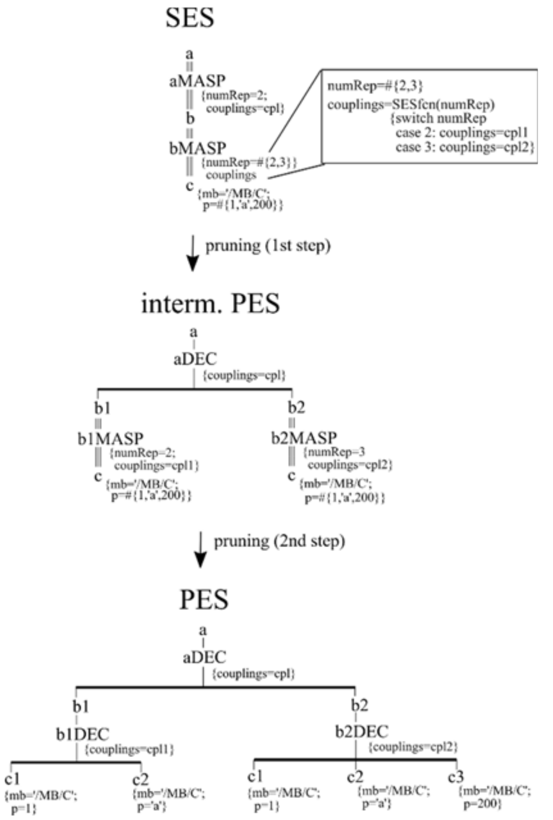


Figure 13: Successive multi-aspects with variable number of replications.

Couplings are set with the help of an SESfcn which chooses cpl1 or cpl2 based on the current value of numRep.

3.5 Design Pattern #14 – Selection Constraints

A modification of pattern #3 (specialization siblings) is shown in Figure 14. With this pattern the use of selection constraints and semantic conditions to limit the possible valid structures is pointed out. The corresponding constructs in feature models are known as *require* and *exclude* [2].

The dotted line from leaf node *c* to node *d* in Figure 14 depicts a selection constraint which means that, if *c* is selected at *a1SPEC*, *d* has to be chosen at *a2SPEC*. The resulting PES is PES3. Note, that current value of VAR2 does not matter for the selection, if VAR1 is set to 2.

Another way to control possible variants are the semantic conditions. The semantic conditions for value combinations of VAR1 and VAR2 interdict the selection of *d*, if *b* is already selected.

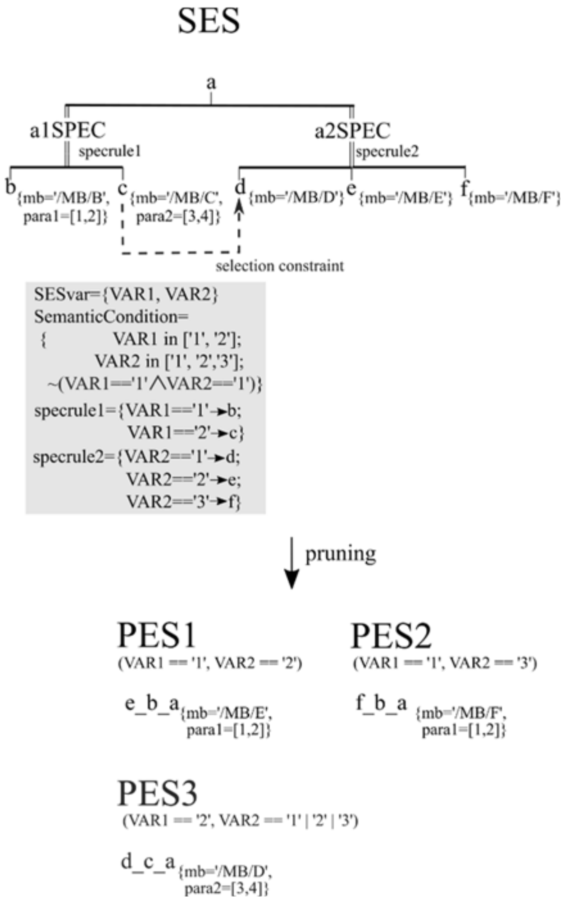


Figure 14: Variant restriction with selection constraints and semantic conditions.

4 Conclusion and Future Work

This paper gives an overview of essential patterns needed to describe and test the pruning process of an SES. With the described patterns, the implementations and tests of the software tools for the extended SES/MB infrastructure developed in the Research Group CEA could be done.

However, while developing and testing the software tools, more combined patterns were found and tested. Some pitfalls in the pruning algorithm were discovered. Therefore, additional attributes, such as the priority attribute, were introduced.

However, e.g. for specialization siblings, there has to be found a way to decide which sibling is evaluated first. Formalized definitions of pruning algorithms are essential part of future work.

Future works regarding the SES software tools will especially cover enhancements of the model builder. For a full software support of the SES/MB-based infrastructure, it is necessary to have a fully functional model builder, which supports the generation of hybrid models with basic systems that are modeled using different paradigms. In addition, the Python implementation will allow to support a set of model builders for different simulators.

References

[1] Deatcu, C., Folkerts, H., Pawletta, T., and Durak, U.. *Design Patterns for Variability Modeling Using SES Ontology*. SpringSim-Mod4Sim 2018, April 15-18 2018, Baltimore, MD, USA, Society for Modeling & Simulation International (SCS), pp. 528-539. (DOI: 10.22360/SpringSim.2018.Mod4Sim.004)

[2] Capilla, R., J. Bosch, and K. C. Kang (Editors). *Systems and Software Variability Management*. Springer, Berlin Heidelberg, Germany, 2013.

[3] Kang, K. C., S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson. *Feature-Oriented Domain Analysis (FODA) Feasibility Study*. Technical Report, Software Engineering Institute Carnegie Mellon University, Pittsburgh, PA, USA, 1990.

[4] Capilla, R., and J. Bosch. *Binding Time and Evolution*. In Systems and Software Variability Management, edited by R. Capilla, J. Bosch, and K. C. Kang, pp. 57-73. Springer, Berlin Heidelberg, Germany, 2013.

[5] Zeigler, B. P.. *Multifaceted Modelling and Discrete Event Simulation*. Cambridge, Academic Press, 1984.

[6] Zeigler, B. P., T. G. Kim, and H. Praehofer. *Theory of Modeling and Simulation*. 2nd ed. San Diego, CA, USA, Academic Press, 2000.

[7] Rozenblit, J. W., and B. P. Zeigler. *Representing and Constructing System Specifications Using the System Entity Structure Concepts*. Proceedings of the 1993 Winter Simulation Conference, pp. 604-611.

[8] Zeigler, B. P., and P. E. Hammonds. *Modeling and Simulation-Based Data Engineering*. San Diego, CA, USA, Academic Press, 2007.

[9] Zeigler, B. P., and H.S. Sarjoughian. *Guide to Modeling and Simulation of Systems of Systems*. Springer, London, UK, 2013.

[10] Röhl, M., A. M. Uhrmacher. *Composing simulations from XML-specified model components*. Proceedings of the 2006 Winter Simulation Conference, pp. 1083-1090.

[11] Wang, S., G. A. Wainer. *Semantic selection for model composition using SAMSaaS*. Proceedings of the 2015 Spring Simulation Conference - TMS-DEVS, pp. 25-32.

[12] Barros, F. J.. *The Dynamic Structure Discrete Event System Specification Formalism*. SIMULATION: Transactions of The Society for Modeling and Simulation International. 13(1996)1, pp. 35-36.

[13] Zeigler, B. P., and H. Praehofer. *System Theory Challenges in the Simulation of Variable Structure and Intelligent Systems*. Computer Aided Systems Theory — EUROCAST '89. EUROCAST 1989. Lecture Notes in Computer Science, edited by F. Pichler and R. Moreno-Diaz., vol 410, pp. 41-51. Springer, Berlin, Heidelberg, 1990.

[14] Pawletta, T., A. Schmidt, B. P. Zeigler, and U. Durak. *Extended Variability Modeling Using System Entity Structure Ontology within MATLAB/Simulink*. Proceedings of the 2016 Spring Simulation Conference – ANSS, pp. 62–69.

[15] RG CEA. *The SES Toolbox for MATLAB / Simulink Website*. http://www.cea-wismar.de/tbx/SES_Tbx/sesToolboxMain.html. Accessed Nov. 22, 2017.

[16] Schmidt, A., U. Durak, and T. Pawletta. *Model-Based Testing Methodology Using System Entity Structures for MATLAB/Simulink Models*. SIMULATION: Transactions of The Society for Modeling and Simulation International. 92(8), pp. 729-746, 2016.

[17] Folkerts, H., T. Pawletta, C. Deatcu, and U. Durak. *Variability Modeling for Engineering Applications*. Simulation Notes Europe vol. 27(4),2017, pp. 167–176.

[18] RG CEA. *The PDEVS Toolbox for MATLAB Website*. http://www.cea-wismar.de/tbx/DEVS_Tbx/MatlabDEVS_Tbx.html. Accessed Nov. 29, 2017.

SimLack: Simulation-based Optimization and Scheduling of Generic Powder Coating Lines

Lukas Hollenstein[1*], Adrian Lötscher[1], Fabian Luccarini[2]

[1]ZHAW Zurich University of Applied Sciences, IAS Institute of Applied Simulation, Grüental, Postfach, 8820 Wädenswil, Switzerland; *lukas.hollenstein@zhaw.ch

[2]e. Luterbach AG, Mülacher 10, 6024 Hildisrieden, Switzerland

SNE 29(3), 2019, 127 - 132, DOI: 10.11128/sne.29.tn.10483
Received: June 10, 2019 (Selected ASIM SST Hamburg 2018 Postconf. Publ.), Accepted: July 22, 2019
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Powder coating and paint-spray lines are often complex production plants because of many dynamical dependencies, limited buffer space and sequence dependent changeover times. We have developed a generic simulation and optimization platform that enables the engineers to design more performant and energy efficient facilities and the production planners to increase productivity through simulation-based optimization. The simulation environment builds on a generic modelling library that captures all variations of such facilities. 'Executable' models are generated automatically from annotated CAD layouts. As a result, the system smoothly integrates with the engineering process. Once the facility is in use, the fully specified virtual plant is used for simulation-based scheduling, employing a combination of a generic priority-based heuristic and a variant of simulated annealing. We discuss how these two aspects of the system render it an important innovation for the painting line industry and show first results from the scheduling system.

Introduction

Powder coating and paint-spray facilities are among the most important energy consumers in the industry, mainly due to their cure ovens staying in operation all day long. To keep operating costs and environmental impact down, machinery manufacturers, like the company e. Luterbach AG, are aiming to optimize the facility layouts and dimensions to their customers' needs.

Once a powder coating line is installed and productive, the operating company intends to plan their production schedule in order to maximize productivity and minimize operation cost. Such facilities normally rely on a closed loop material handling system that transports the parts to be treated on hangers along chains. Failing to feed the system with an efficient job sequence can lead to a significant drop in throughput or even to deadlocks.

Since a few years, Luterbach is already at the forefront of designing energy efficient facilities through simulating the energy and heat budget by means of pinch analysis [1]. The next step is to quantitatively optimize the design and the usage strategies of such systems with respect to the material flow and production performance. Even though manufacturing resource planning (MRP II) for generic production facilities may exist on the market the specific industry sector (powder coating and paint-spraying) has not embraced such solutions. Plant operators are often not employing enterprise resource planning (ERP) software throughout the production process or, if they do, struggle to enable it with sophisticated scheduling abilities that go beyond simple heuristics based on the planner's experience. In addition, when designing new facilities and retrofits, it appears that most machinery manufacturers still rely on static analysis and experience to plan the facility layouts. These approaches are bound to fail as facilities grow larger, more flexible and complex.

The scheduling problem for generic powder coating and paint-spray facilities can in principle be seen as hybrid flow shop scheduling with unrelated parallel machines [2, 3, 4]. Unfortunately, the rules that govern the dynamics of the facilities can be highly specific and complex, as can be the routes of parts through the system and the timing involved.

As a result, the scheduling problem for such manufacturing systems is not straightforward to formulate in closed form in order to apply mathematical programming techniques or well-known heuristics. Therefore, a simulation-based approach is more promising [5, 6].

Apart from scheduling, there are two more reasons for a simulation approach in the context of plant engineering: first, the facility design phase highly profits from a generic simulation framework than can be used to rapidly build models and compare different scenarios and configurations. Secondly, a dynamic visualization of the system helps to understand it, to pinpoint bottlenecks and, importantly so, supports sales with a quantitative proof of concept. Wilson & Zettle report on an operative scheduling solution for powder coating lines that relies on existing simulation models developed in the planning phase of the facility [7]. However, it is unclear in which way the simulation is used in their heuristic scheduling algorithms and if the scheduling methodology is generically applicable to any facility configuration.

We extend this approach in two ways. Firstly, we specify a generic annotation scheme for computer aided design (CAD) layouts that enables automatic model generation for powder coating lines. Secondly, we discuss a scheduling methodology that combines a generic priority-based sorting heuristic with a variant of the simulated annealing meta-heuristic that improves the solution by means of many simulation runs. We have implemented these concepts as an integrated platform for simulation-based optimization and scheduling of powder coating and paint-spray facilities. The simulation environment is called *SimLack*, derived from *Simulation* and *Lackieranlagen*, the German word for paint shop.

1 Automatic Model Generation from CAD Layouts

When planning new powder coating and paint-spray facilities or designing a retrofit, machinery manufacturers aim at optimizing the facility layout and number and dimensions of building blocks in order to minimize the energy consumption and maximize the total throughput. Modelling and simulations enable prototyping, iterative refinement and quantitative analysis of designs.

However, firstly, building and analyzing simulations is time consuming and, secondly, the engineers are usually no simulation experts.

To overcome this issue, we have developed an integrated simulation environment that consists of a generic modelling library for painting lines with an associated domain specific language based on annotated CAD drawings to generate simulation models automatically. A discrete-event simulation engine runs the models while showing the model state as a schematic visualization auto-derived from the original CAD drawing.

1.1 The generic model

Painting line facilities all follow similar principles and consist of the same building blocks, but vary in their details. A hanging conveyor system of some kind (often a *Power and Free* chain) is carrying the parts to be treated in hangers through a series of processing steps including pre-treatment, powder application or paint-spraying, curing, and cooling. Other intermediate steps as well as buffering and inner closed loops can be present, which complicate the system. Due to space and cost restrictions, the lines typically cannot be kept lean such that material and color-dependent changeovers have to be taken into account and dynamic dependencies appear at certain points in the facility.

The generic discrete-event model describes each job as a number of parts on a hanger. The hangers are the entities that are moved through the system. The system is defined by the *facility configuration* that consists of

- the *positions* at which the hangers can reside,
- the *segments* of the hanging conveyor network connecting the positions,
- the *decision points* where two segments join or separate, and
- the *processes* in which the hangers can change state, representing either a loading point (source), a generic process (pretreatment, powder application, curing, cooling, etc.), or a discharging point (sink).

Each position belongs to exactly one process. Each segment consists of a sequence of positions. The segments must form a closed loop globally.

Next to the facility configuration, the model is specified by the following additional data:

- the sequence of processes a job has to pass through (incl. processing time or minimal/maximal retention time),
- specialized prioritization rules per decision point (optional), and
- sequence dependent changeover times per process (optional),
- availability patterns of processes to represent shifts and planned outages (optional).

The entities (hangers) are moved from one position to the next along the segments and cannot overtake other entities. Positions can have capacity larger than one to allow for quickly adding buffer space in experiments. The routing of entities through the system is derived from the job's sequence of processes and the prioritization rules at decision points that can depend on the job attributes and the state of processes.

The model is generic in the sense that it is agnostic about the actual meaning of the processes and even the attributes of the jobs that influence the processing time, the changeover times and the prioritization rules in decision points. The semantic behind the processes and the job attributes is only in the data that is provided to generate the simulation model. This means the model represents very generically any kind of closed-loop flow shop system. It is therefore widely applicable, not only to the partnering company and not even only to painting lines.

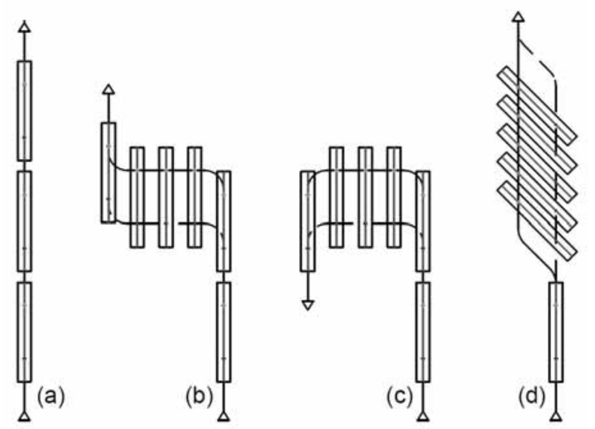


Figure 1: Example snippets from a CAD layout. Segment (a) has three positions in longitudinal direction; (b) is a transversal buffer where hangers exit in the same direction as they enter, i.e. no change of orientation; (c) is a transversal buffer with change of direction; and (d) is a diagonal buffer without change of direction.

1.2 The CAD-based domain specific language

In order to generate executable instances of the generic simulation model the complete facility configuration incl. its data needs to be provided. Machinery engineers use CAD drawings when designing new plant layouts or investigating changes to existing facilities. To integrate the simulation library seamlessly into the workflow of the engineers we devised an annotation scheme for CAD layouts. The annotated CAD layout is imported into the simulation environment and turned into a specific simulation model automatically. The data needed to fully specify the simulation experiment (jobs, prioritization rules, changeover times, and availabilities) is not provided through the CAD-interface and needs to be imported separately. Here we focus on the CAD-based domain specific language, as it is the main innovation that enables automatic model generation.

CAD layouts of painting lines are complex and contain many elements, most of which do not provide information for generating the model. Fig. 1 shows snippets from a CAD layout exemplifying sequences of positions (i.e. segments). A drawing does not naturally contain semantic information, e.g. even though positions are always rectangular shapes it is not clear from the outset which process the positions belong to. Therefore, we require the elements to be placed in blocks (aka layers) that are named semantically and structured hierarchically according to the following scheme:

- Block `Processes`: contains all processes, e.g.
  - Block `LoadingPoint`: contains all its positions, e.g.
    - Block `P001` (with rectangle)
    - Block `P002` (…)
  - Block `Buffer1`: e.g.
    - Block `P003` (…)
    - …
    - Block `P010` (…)
  - …
- Block `Segments`: contains all segments, e.g.
  - Block `P000-P002_0` (with lines and arcs)
  - Block `P002-P010_1` (…)
  - …

There are optional naming conventions for the processes such that the type of process can be recognized (loading point, genric process, or discharging point). The positions can be named arbitrarily but uniquely. In the example above we applied the optional convention to name the positions `P001` … `P010`.

In order for the importer to uniquely identify the starting and ending positions of segments, we require the following convention: the names of segments must be made up of the *last position of the preceding segment* and the *last position of the current segment*. For example the segment `P000-P002_0` starts at the position `P000`, that is the end of the preceding segment, and ends at position `P002`. The importer algorithm identifies the intermediate positions along the segment geometrically.

Segments feature another complication, namely the possibility that the orientation of the hangers can change in the end of a transverse buffer, see Fig. 1. Because such situations are not straightforward to detect algorithmically we require the name of the segment to end either with `_0` for “no change of orientation” or with `_1` for “change of orientation.”

The combination of CAD drawing and annotation scheme (naming convention) can be seen as a domain specific language that enables data-driven model generation. Because the annotation with specifically named blocks is lightweight in its application, this approach integrates neatly into the workflow of the plant engineers. Other possibilities such as specifying the facility configuration in tables or using a graphical drag-and-drop editor would certainly lead to significant overhead in the design cycles. The specification is domain specific in the sense that it does rely on consistency relations that hold specifically in the powder coating domain, even though it may be applicable more generally to other closed-loop hybrid flow shop systems.

2 Simulation-based Optimization

Production planners of painting lines currently use experience-based heuristics and simple static estimates to put together production schedules given a number of orders.

Algorithmic assistance is highly asked for; however, acceptance of decision support systems strongly depends on the level of control and understanding the user can have.

With this requirement in mind, we have developed a hybrid scheduling scheme that builds on the user’s planning experience and knowledge of the facility and product characteristics. The method works in two phases that we describe briefly in the following.

2.1 Priority-based heuristic

- The user selects a number of *sorting priorities*. These are job attributes like material, paint color, due date, etc. and can be configured freely for each specific facility.
- All permutations of the sorting priorities are generated.
- Each permutation is then used to generate a job sequence by sorting hierarchically according to that sequence of priorities.
- For n selected attributes this procedure yields $n!$ priority-sorted job sequences.
- All sequences are evaluated using the simulation model of the facility, respecting all constraints like shift plans, changeover times, outages, etc.

The result is twofold: A good estimate of the optimal schedule (according to some objective function like the makespan) and insight into which job attributes have a strong impact on the performance of the job sequence. This insight helps to improve acceptance by the user and the user can explore and learn about the facility.

2.2 Simulated annealing for refinement

Starting from the best job sequence found by the priority-based heuristic, we employ a variant of *simulated annealing*. This class of stochastic meta-heuristics is well-known to be able to find globally near-optimal solutions with certainty and has proven convergence properties [6]. The basic algorithm works as follows:

- In each iteration a random solution in the neighborhood of the current solution is generated.
- The objective function f is evaluated and if the new solution is better than the current one it is accepted directly as the next step.

- If the new solution is not better, it is accepted all the same with a probability proportional to $e^{-\Delta f/\theta_i}$. Here Δf is the positive difference between the new and the current objective value. The parameter θ_i is a positive number that depends on the iteration i. It starts relatively large which means the probability of accepting poor solutions is large and therefore exploration of the solution space is made possible. With increasing number of iterations θ_i is decreasing such that poor solutions are only seldom accepted, exploration ceases and exploitation (local refinement) is emphasized.
- As soon as the globally best solution does not improve anymore for some number of iterations or the maximal number of iterations is reached the algorithm stops.

The difficulty in applying simulated annealing is to define an efficient solution neighborhood, i.e. a procedure for generating new solutions nearby a given solution. A simple choice would be to swap a random single pair of adjacent jobs in the sequence, *single adjacent interchange*. We compared the performance of a number of different neighborhoods and found that the scheduling problem at hand asks for more exploration than single interchanges. Therefore, we make the neighborhood depend on the iteration: the number of adjacent interchanges starts at the length of the job sequence, i.e. large exploration, and subsequently reduces to one with increasing number of iteration. A similar effect could be achieved with single adjacent interchange and θ_i reducing more slowly, but we found this variant to converge more quickly for a set of reference problems.

2.3 Preliminary results

Figure 2 compares random sampling of the solution space with the proposed probability-based heuristic and simulated annealing. We show the distribution of objective values from the evaluated job sequences. In this case, the objective function is the makespan that is minimized. The distributions is multi-modal (three clusters appear) which is due to the discontinuity of the objective function. Observe that random sampling is not able to find the best-performing solutions with makespan below 160 h. The priority-based heuristic generates a broad range of solutions incl. some that come close to the best ones found.

The simulated annealing algorithm, however, is able to improve on the soring heuristic and spends most of its time near the optimum trying to improve on it locally. These are preliminary explorations and more systematic performance analyses and comparisons are left for future work.

3 Summary and Outlook

The integrated simulation and optimization environment *SimLack* essentially provides two important innovations to the industry sector of powder coating and paint-spray lines. Firstly, a generic modelling library and a domain specific language were developed to provide the possibility to generate simulation models automatically from CAD layouts. This approach integrates the simulation environment seamlessly into the workflow of the plant engineers. Secondly, a simulationbased scheduling methodology was devised, based on a combination of a generic sorting heuristic and a variant of the simulated annealing metaheuristic.

After importing a CAD layout into *SimLack*, a graphical user interface eases tweaking the facility configuration, data import and editing, the specification of detailed prioritization rules to apply at decision points, setting up changeover criteria and durations, and shift plans. The integrated platform enables the plant engineers to set up, carry out and organize simulation experiments to compare different facility layouts and various parameter settings.

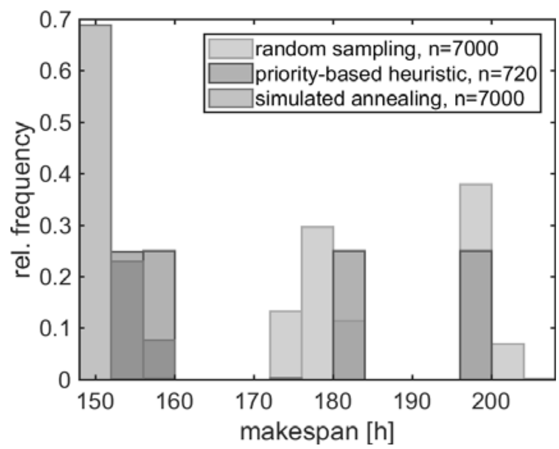


Figure 2: Comparison of different algorithms in terms of the distribution of the objective values (makespan) from evaluated job sequencies in the course of the optimization runs.

A first use-case shows that e. Luterbach AG was able to acquire a new contract successfully through the ability to fine-tune the plant details and to convince the customer with quantitative estimates of its performance. In this case, the seamless integration of the *SimLack* platform proved crucial for its success, as the engineers went through several iterations to refine the layout to the point where requested performance could be achieved.

Preliminary performance tests of the scheduling system with small and medium sized problems show:

- Even though the sorting heuristic in the first stage can provide relatively good solutions the second stage can typically improve on it in a few thousand iterations (simulation runs), depending on problem size.
- Our simulated annealing variant with a neighborhood with decreasing number of adjacent inter changes is converging more efficiently than variants with a constant number of adjacent interchanges.
- Since simulations are run in parallel, a scheduling run for a large sequence of 790 jobs on a normal sized facility takes about 30 min (for 7000 simulations) on a quad-core laptop to run.
- These first results are encouraging to refine details of the *SimLack* system and its algorithms. Next to more in depth performance analysis, we are planning to add alternative algorithms that the user can choose from and compare against each other. Further, a challenging improvement is to add support for the plant design process by providing means of automatically optimize certain parameters and aspects of a facility configuration.

Acknowledgement

We thank Florian Eggenberger and Ramon Kramis from e. Luterbach AG and Ivo Berchtold and Othmar Winiger from wirth+co AG for their constructive collaboration and the two companies for their support. This work is supported by the Commission for Technology and Innovation (CTI) of the Swiss Confederation through grant number 17655.1 PFEN-ES.

References

[1] ke. A quantum leap. *IST International Surface Technology*. 2017; 10(1):16–21.

[2] Pinedo ML. *Planning and Scheduling in Manufacturing and Services*. New York: Springer-Verlag, 2nd ed. 2009.

[3] Gupta JND, Stafford EF. Flowshop scheduling research after five decades. *European Journal of Operational Research*. 2006;169(3):699–711.

[4] Ribas I, Leisten R, Framiñan JM. Review and classification of hybrid flow shop scheduling problems from a production system and a solutions procedure perspective. *Computers & Operations Research*. 2010; 37(8):1439–1454.

[5] Kiran AS. Simulation and Scheduling. In: *Handbook of Simulation*, edited by Banks J, This chapter contains sections titled: Introduction, Definitions and Background, Scheduling Approaches, Simulation of Job Shops, Simulation-Based Scheduling: Application, Implementation Issues, Concluding Remarks, References, pp. 677–717. Wiley-Blackwell. 2007.

[6] Gosavi A. *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. Operations Research/Computer Science Interfaces Series. Springer US, 2nd ed. 2015.

[7] Wilson H, Zettl R. Simulation and Optimization Based Flexible Job Scheduling of Powder Coating Lines. In: *Digital Product and Process Development Systems*, edited by Kovács GL, Kochan D, IFIP Advances in Information and Communication Technology. Springer, Berlin, Heidelberg. 2013; pp. 369–375.

oHMint: An Online Mathematics Course and Learning Platform for MINT Students

Helena Barbas[1], Franz Konieczny[2], Alexander Lohse[1], Thomas Schramm[3*]

[1]Universität Hamburg, Fachbereich Mathematik, Bundesstraße 55, 20146 Hamburg, Germany

[2]integral-learning GmbH, Clausewitzstraße 2, 10629 Berlin, Germany

[3]HafenCity Univ. Hamburg, Überseeallee 16, 20457 Hamburg, Germany; **thomas.schramm@hcu-hamburg.de*

SNE 29(3), 2019, 133 - 136, DOI: 10.11128/sne.29.en.10484
Received: June 15, 2019 (Selected ASIM SST Hamburg 2018 Postconf. Publ.), Accepted: July 22, 2019
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. The oHMint project is an initiative with the goal of providing an online course and learning platform for STEM students of higher mathematics. Its flexible design allows it to be used for self-study as well as blended learning scenarios including flipped classrooms. In a pilot project in 2017/18 the chapter *Differential Calculus* is being produced as a prototype of an oHMint unit. This pilot is organized and funded through the Hamburg Open Online University, with technical support and implementation by integral-learning GmbH. It is intended as a catalyst for the future development of the full oHMint course spanning four semesters. A unique characteristic of oHMint is its broad supporting base among a variety of German institutions in the form of the OMB+ consortium.

Introduction

Engineering mathematics is the main tool to describe the world and to simulate its processes. A profound mathematical education is therefore essential for future engineers and scientists. Digitalization is becoming increasingly important in teaching and studying at university level. More and more online learning tools are being developed to address the current need for educational resources that go well beyond the classical textbook by providing high temporal and spatial flexibility for users while at the same time offering an interactive learning environment.

In this spirit, the oHMint project was initiated through the OMB+ consortium, a group of 14 German institutions of higher education. The acronym stands for **o**nline **H**igher **M**athematics (for m)**int**.

It is the aim of oHMint to provide university students of STEM degree programmes with a modern way of transitioning smoothly from a high school level of mathematics to a bachelor degree level. The OMB+ consortium has already developed the online mathematics bridging course OMB+, which is widely used and recommended as a preparation for those interested in a STEM university degree programme. This experience is a valuable asset for oHMint.

Currently, in a pilot project with a lifespan of 14 months one base unit of oHMint is being created. This serves as an opportunity to test new didactical approaches for mathematics online courses as well as innovative types of exercises addressing the specific needs of the current student generation.

1 Background of oHMint

1.1 OMB+ mathematics bridging course

The OMB+ consortium has developed and maintains the well-accepted "Online Mathematics Bridging Course Plus" OMB+ [1] under the auspices of the TU9, a group of nine important German technical universities. The course is based on the recommended cosh catalogue [2], which sets a standard of mathematical requirements for freshmen in STEM studies, and should also give an orientation for school teachers in mathematics. The OMB+ is text oriented, but includes a large amount of questions, interactive elements, videos and examples with standard solutions, which can be uncovered systematically.

As of today, around 50 German institutions use and recommend the OMB+, which is available in German and English and currently being translated into Chinese. Supplementary chapters covering e.g. stochastics, complex numbers and formal logic have been added and further educational videos are in the making.

The oHMint project uses the same enriched text oriented style and shares similar features. For more information on OMB+ see [1] and [3].

1.2 The Hamburg Open Online University

The acronym HOOU stands for Hamburg Open Online University (www.hoou.de), a joint initiative of six institutions of higher education in Hamburg: Universität Hamburg (UHH), Hamburg University of Applied Sciences (HAW), Hamburg University of Technology (TUHH), HafenCity University Hamburg (HCU), Hochschule für bildende Künste (HFBK) and the Hochschule für Musik und Theater (HFMT). The Senate of Hamburg founded it in 2015 as the educational branch of “Strategie Digitale Stadt” (Strategy Digital City), an initiative to bundle processes of digitalization within the city.

The HOOU focuses on learners and collaboration for problem solving and develops study material at academic level. Its target groups are not only the custom university students but also interested non-academic persons. Thus, all study material is open as Open Educational Resource and is distributed under a CC license. The production of the first chapter of oHMint is funded by the HOOU and running from November 2017 through December 2018.

2 Modularisation and Scope

2.1 Course structure

The OMB+ consortium designed the preliminary structure of the curriculum for the full oHMint course. It spans four semesters and aims to cover all the mathematics typically taught in STEM studies at German institutions of higher education. The curriculum was carefully structured, keeping in mind that different (types of) universities have different needs for their respective degree programmes. The course will be available in English and German. With oHMint we aim to create a course that consists of units which can be combined in various ways to cover these needs. The fact that a broad spectrum of institutions is represented in the OMB+ consortium justifies our hope that the final course will be widely accepted within the relevant community. This clearly distinguishes oHMint from existing higher mathematics online courses in Germany.

There are three types of units in oHMint: base units cover standard content and are expected to be included in every higher mathematics course, regardless of the institution it is taught at. Supplementary units go deeper into the (theoretical) background of a base unit, and include more abstract views on base unit concepts as well as more involved parts. Finally, optional units exist for subjects that are not necessarily always part of a higher mathematics course, depending in particular on the study programme and/or institution. The idea is that lecturers will be able to choose and combine the units that are relevant for their classes to provide students with everything they need (but not more).

Parallel to the development of the content of oHMint, a data base is being created that keeps track of all the interconnections between mathematical concepts and results in the units and which will help to ensure the desired modularity of the course.

The units themselves are text-based and consist of lectures, exercises, trainings and final tests. They are supplemented by videos where important concepts or ideas are also shown. Each chapter starts with a motivation for the new mathematical concepts taught such that students develop a sense for the importance of the content they are supposed to learn. Moreover, many interactive elements are being implemented in oHMint, some are discussed in more detail below.

2.2 Content of the first-semester course

The learning material from which a lecturer can choose for a first-semester oHMint course consists of:

- Three base units: numbers and functions, differential calculus, integral calculus.
- Four supplementary units: differential calculus, integral calculus, sequences, series.
- Eight optional units: continuity, determination of zeros, L’Hospital’s rule, partial fraction decomposition, approximate integration, sequences, series, Fourier analysis.

The scope of the content of each unit has been roughly assessed in terms of the European Credit Transfer and Accumulation System (ECTS) which is commonly used at European institutions of higher education. ECTS points measure the volume of learning based on workload and desired learning outcome. In Germany, one ECTS point is equivalent to 25-30 hours of studying.

The oHMint working group has made an assignment between the content of the chapters of the first-semester course and ECTS points. The base units are assigned 1 ECTS point each, the supplementary units and optional units are assigned 0.5 ECTS. The workload for the corresponding units is estimated to be within the given range.

These values are used as an orientation for the lecturers who use the material for their courses. The actual assignment of credit points to the course is made by the lecturer or their institution.

3 Didactical Concepts of oHMint

3.1 Didactical methods

The oHMint materials are designed to be used for a wide range of blended learning environments up to fully web based self-learning courses. We selected our didactical methods to specifically suit these environments.

Our guiding principles are a spiral curriculum (after J. Bruner, see [4]), the development of intuitive understanding, awareness, and knowledge of mathematical thinking (see e.g. [5], [6], [7]), and in general an application oriented approach. We follow the ideas of cognitive load theory [8] which focuses on the limited capacity of working memory of students. New ideas and concepts are introduced incrementally, always keeping the cognitive load small and focused. The formally exact representation is the result of the teaching process and not as often a starting point, making the constructive process of mathematical knowledge visible to the students and fostering the awareness between intuitive thinking and correct formulation of mathematics.

Our database of mathematical interconnections (see Section 2.1) supports the development of the spiral curriculum as examples and exercises are shared between the units and viewed from different perspectives.

The revised version of Bloom's taxonomy [6] (see also [15]) is used for a careful evaluation of the learning goals for each unit and exercises and quizzes are designed to reflect this information. This allows the students to assess their learning achievements and the growth of their knowledge continuously. Automatically corrected problems on different levels of content provide them with a differentiated feedback to support the learning process. Different types of exercises support the comprehension of ideas, the stabilization of acquired knowledge or the development of routine skills. Application-oriented exercises from science and engineering impart an understanding of the importance of the methods to the relevant application domains.

3.2 Serious games and gamification

Serious games are games that are designed in our context for educational purposes. Gamification can then be defined as the use of typical game design elements and scenarios in non-game contexts, see [9]. We develop a variety of such components for oHMint which create a new incentive for active participation. It is the aim of the current pilot project to test several approaches and choose the most successful ones for implementation in the entire course which will subsequently be developed. Some of our components are:

Badges: We are developing a system of badges that participants receive for completing certain tasks. The positive effect of badges has been documented, see [10]. This will however be an optional element since not everyone likes to be “distracted” by such matters.

Exercises in game form (competitive and non-competitive): For skills that require routine we are developing a gaming approach. The player is given a task such as “what is the derivative of the function f(x)”. If answered correctly, the player receives points and proceeds. After finishing the game, players can enter their score into a high score list. We are also investigating the option of enabling players to compete against each other in a duel or small groups and the possibility of a “Who wants to be a millionaire?”-styled game. Developing and realizing these ideas and more general the whole course we benefit from close interaction with the e-learning group at Universität Hamburg as well as with student helpers also from the HafenCity University who provide valuable feedback from the target group of oHMint.

3.3 The advantage of the oHMint concept

Because of the modularisation (see 2.1) lecturers can design their own course by selecting units depending on their programme-level.

Innovative types of exercises are implemented: audio files train the transition between written and spoken mathematics; in “reversed” exercises the students shall find flaws in a given line of arguments; in “drawing” exercises they are asked to sketch the graph of a function with certain properties. Moreover, there is a large amount of quick checks, i.e. short questions or problems interspersed in the course where students can immediately check if they understood some new content or not.

In the available group instructing mode professors can not only compile a course suitable for their needs but also follow the learning progress of their students in a gradebook overview. Also, a flipped classroom version will be available (see 4). And last, a call centre run by integral-learning GmbH will be integrated to oHMint as it is already in the OMB+. Thus, participants of oHMint can contact a call centre for mathematical questions from 10 am till 8 pm via internal chat, telephone, skype (telephone and chat) or internal forum as a free service. For more details on our didactical methods see [11].

4 Flipped Classroom

At many universities a typical mathematics lecture for engineers suffers from low intrinsic motivation of the students, their low competence in school mathematics, (which we hope to enhance through OMB+) and the procrastinating study style of many students.

Flipped classroom is an interactive teaching model that addresses this problem. The information and content instruction takes place outside of class time and class time is used for practice oriented activities. The concept has been successfully introduced in university mathematics and evaluated in recent years [12, 13, 14]. It has been shown that the average performance of students in exams is at least equal to the one of students instructed with traditional methods. Moreover, through flipped classroom students tend to have a better understanding of underlying concepts. The gap between better and weaker students tends to get smaller while the well performing students perform equally well as or better than with traditional classroom lectures.

The students get preparatory material focusing on the basic concepts with a set of key questions guiding their studies and a quiz of control questions that have to be answered before class time. The quizzes are automatically corrected and give students helpful feedback and encourage them to prepare.

In-class time is used for interactivity with more challenging problems that are followed up with homework.

In winter term 2018/19 the first prototype module for differential calculus will be used in a flipped classroom context for a freshmen class of Mathematics in a Geodesy and Geoinformation course at the HCU. The class time will be split. The first half will be an interactive session to handle feedback questions from students and presentations from students. The second half will be used for group work on special exercises to solidify and deepen the gained new mathematical competencies.

Implementing the first base unit of oHMint into a real teaching scenario at this early stage will result in valuable experience and feedback for the further development of the oHMint course.

5 Conclusions

As explained above, the production of the first oHMint base unit *Differential Calculus* is funded by the HOOU. It is meant as a sample for the remaining units of the course, testing new approaches in digital teaching and innovative types of exercises. The oHMint working group, currently a subgroup of the OMB+ consortium, intends to create attention for the project and raise further funding through this sample unit. Other working groups are encouraged to join the oHMint project and/or produce an oHMint unit themselves.

The main assets of oHMint are its flexibility in terms of teaching style, ranging from self-study to flipped classroom settings. Each unit will be translated into English once it is completed. A preliminary version of *Differential Calculus* will be put to the test in the winter term 2018/19 at HCU Hamburg as a flipped classroom version, feedback from HCU students and lecturers will be incorporated into the future development of oHMint.

Overall, we believe that oHMint poses an important step in the digitalization of STEM education in Germany. The broad supporting base guiding its development ensures that it is tailor-made for the needs of lecturers and students alike, and its modular structure makes it suitable for all institutions.

Acknowledgements. We gratefully acknowledge the financial support of the HOOU for the current project, the guidance and input of the OMB+ consortium (especially Volker Bach) in the development of oHMint and support beyond the technical lead by integral-learning GmbH.

References

[1] OMB+ www.ombplus.de

[2] Braun I, Schröder J, et al. *cosh cooperation Schule: Hochschule – Mindestanforderungskatalog Mathematik (Version 2.0)*, 2014. https://lehrerfortbildung- bw.de/u_ matnatech/mathematik/bs/bk/cosh/katalog/in- dex.html

[3] Barbas H, Schramm T. The Hamburg Online Math Test MINTFIT for Prospective Students of STEM Degree Programmes, *MSOR Connections*, 2018, 16(3): 43-51.

[4] Bruner J. *The Process of Education, Harvard University Press.* Cambridge, Mass., USA, 1960.

[5] Pintrich P R. The Role of Metacognitive Knowledge in Learning, Teaching and Assessing, *Theory into Practice*, 2002, 41(4): Revising Bloom’s Taxonomy, 219-225.

[6] Krathwohl D R. *A Revision of Bloom’s Taxonomy: An Overview*, 2002. https://www.depauw.edu/files/resources/krathwohl.pdf

[7] Zech F. *Grundkurs Mathematikdidaktik*, Beltz Verlag, Weinheim und Basel, 1996.

[8] de Jong T. Cognitive load theory, educational research, and instructional design: some food for thought, *Instructional Science*, 2010, 38: 105-134.

[9] Deterding S, Dixon D, Khaled R, Nacke L E. Gamification: Toward a definition, *CHI 2011*, 2011, Vancouver, Canada.

[10] Denny P. The Effect of Virtual Achievements on Student Engagement, *CHI 2013: Changing Perspectives*, 2013, Paris, France, pp. 763-772.

[11] Barbas H, Gasser I, Konieczny F, Lohse A, Seiler R. oHMint: Online Higher Mathematics for MINT Students - An Online Mathematics Course and Learning Platform, Preprint, to appear in the Proceedings of the *46th SEFI Annual Conference,* 2018, Copenhagen, Denmark.

[12] Epstein. *The Calculus Concept Inventory -- Measurement of the Effect of Teaching Methodology in Mathematics*, Notices of the AMS 2013, 60(8), https://www.ams.org/journals/notices/201308/rnoti- p1018.pdf

[13] Maciejewski W. *Flipping the calculus classroom: an evaluative study*, Teaching Mathematics and its Applications: An International Journal of the IMA, Volume 35, Issue 4, 1 December 2016, Pages 187–201, https://doi.org/10.1093/teamat/hrv019

[14] Cronhjort M, Filipsson L. Weurlander M. *Improved engagement and learning in flipped-classroom calculus*, Teaching Mathematics and its Applications: An International Journal of the IMA, , hrx007, https://doi.org/10.1093/teamat/hrx007

[15] Anderson L W, Krathwohl D. (eds.) *A Taxonomy for Learning, Teaching, and Assessing: a Revision of Bloom's Taxonomy of Educational Objectives.* New York Longman, 2001.

Influence of TAS' Characteristics on the Related Drayage Network

Ann-Kathrin Lange*, Kristof Ole Kühl, Anne Kathrina Schwientek, Carlos Jahn

Institute of Maritime Logistics, Hamburg University of Technology,
Am Schwarzenberg-Campus 4, 21073 Hamburg; *ann-kathrin.lange@tuhh.de

SNE 29(3), 2019, 137 - 140, DOI: 10.11128/sne.29.sn.10486
Received: June 15, 2019 (Selected ASIM SST Hamburg 2018 Postconf. Publ.), Accepted: July 22, 2019
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Truck drayage transports in the port connect container terminals with other logistics nodes as empty container depots, freight stations or customs stations. Due to the large proportion of drayage transports in the overall truck arrivals at container terminals and their relatively high costs in the maritime transport chain, drayage transports have high importance in port processes. To reduce peaks in truck arrivals, container terminals more and more often implement truck appointment systems (TAS), which require trucking companies to book specific time windows for handling prior to their transports. Besides on their impact on the container terminals, these TAS also effect the other stakeholders in the drayage net- work, which has been neglected in scientific studies so far. This study aims to analyze the effect of TAS capacity and utilization on the arrival times at other logistics nodes as well as on the number of successfully executed orders per truck.

Introduction

Containerized maritime transport grows continuously since 2010 as well as before 2009 [1]. To enable lower prices and to diminish the emissions per transported container, shipping companies are ordering larger and larger vessel sizes. In 2006, the largest ship by far was the Emma Maersk with a capacity of about 15,000 Twenty-Foot equivalent Units (TEU). In contrast, the largest ship in 2017, the OOCL Hong Kong, has a capacity of over 21,000 TEU. The growing vessel sizes pose many different challenges for the container terminals as well as for the overall port area. When considering the landside operations of a container terminal, the main challenge is the peaks in truck arrivals causing the terminal gate to have either too few personnel to prevent long queues reaching the access roads to the terminal or too much personal causing high costs for the terminal. Furthermore, the waiting trucks lead to high CO_2-emissions due to their running engines. To mitigate these effects, an effective solution is to implement a truck appointment system (TAS) at the container terminal. A TAS is a vehicle booking system used to control the number of trucks arriving at the terminal at different times of the day. With this system, the trucking companies have to book specific time windows to deliver or pick up a container at the terminal. This does not only affect the container terminals and the trucking companies, but also other operative companies in the port, due to shifts in truck arrivals and higher restrictions in the dispatching process of trucking companies.

The aim of this study is to analyze the effects of varying TAS' characteristics on different operative stakeholders in the port network. To do so, many different characteristics, e.g. the length of the time window or the capacity per time window, are studied on their singular and combined impact on the arrival times at logistics nodes as well as on the amount of successfully executed orders per truck. First of all, the state of research is presented. Afterwards the simulation study and the experimental design are described. Finally, the results of the study are outlined and a conclusion is given.

1 Port Drayage Operations

Port drayage is defined as "truck pickup from or delivery to a seaport, with the trip origin and destination in the same urban area" ([2], [3]). Sometimes, drayage transports are also called inter-terminal transports (ITT), with the one difference, that ITT always take place between different container handling areas in one port and does not consider other logistics nodes as freight or customs stations. The main cause for drayage transports is the necessity to transport shipment containers from one terminal to the next. On container terminals, there are mainly three types of orders: import, export and transhipment. Import container arrive at the terminals via ocean carrier and leave on barge, train or truck. Export containers arrive vice versa per barge, truck or train and leave per ocean carrier. Transhipment containers are discharged from one vessel, stored, and afterwards loaded on another vessel. As not all vessels stop at all container terminals in one port, many transhipment containers need to be trans- ported to other terminals, mainly by truck and some- times by train or barge. Furthermore, empty containers, which often have long storage times, are mainly transported to empty container depots to save storage space on seaport container terminals. Other containers are packed or unpacked in freight stations or need to be transported to customs stations. All these transports are called drayage transports if the origin and destination are in the same area. Due to their large proportion of truck arrivals at container terminals and their relatively high costs, they present an important part of the overall maritime supply chain (inter alia [4], [5]).

2 State of Research

The first TAS was introduced in the ports of Los Angeles and Long Beach in 2002 in response to California Assembly Bill (AB) 2650. The evaluation of the program was mixed, due to high entry barriers for the truck drivers, as varying systems for registration and a generally high effort for the overall process, and therefore due to a low participation [6]. Main reason for the intro- duction of a TAS is the need to reduce CO_2-emissions. Other goals are to reduce truck waiting times at the gates or to improve the terminal processes. Therefore, the TAS as well as other approaches, as webcams at the terminal gates to provide information about the current congestion (e.g. [3]) or to promote transports outside peak times by introducing varying tolls in the port (e.g. [7]), have been studied increasingly. Today, several successful TAS are running in different parts of the world, e.g. Vancouver, Sydney and Southampton, but the development goes on to improve these systems or to find better alternatives (inter alia [8], [9]).

Scientific studies focus mainly on the effects of TAS on container terminal productivity (e.g. [10], [11]). Other publications also consider the possible benefits and challenges for trucking companies (e.g. [12], [13]). To the authors' knowledge, other actors, as empty container depots, freight stations or customs station, are never considered. For a comprehensive overview about literature on port drayage transports and TAS be referred to Lange et al. 2017 [14].

3 Simulation Study

As described above, the focus of this study is on port drayage transports. Therefore, only transports in the port area and between the relevant operative stakeholders are considered. For the simulation study, the program Tecnomatix Plant Simulation 13 is used because it is widely recognized in industry and this research area. The simulation model is generated based on operative data of different stakeholders in the port of Hamburg, especially transport data from trucking companies and process durations from various logistics nodes. A simulation run covers one workday from 0 to 24 o'clock. Every simulation experiment is repeated 20 times.

For the simulation model, a flexible list of orders for one considered trucking company is generated. In this list, the source and the drain of every order is noted. In the next step, time windows are booked for all orders with either source, drain or both at container terminals. For all time windows specific probabilities for a successful booking are considered, based on the assumed utilization of this time window by other trucking companies. The order list is imported in the simulation model. There, all relevant stakeholders are displayed. In every simulation run, one trucking company, four container terminals, six empty container depots, six freight stations and five other logistics nodes are considered. The driving distances between the individual stakeholders are represented by a distance relation matrix, considering the driving durations at different times of the day due to traffic. The durations are determined by using a Google Maps API for the relevant routes between logistics nodes in the port of Hamburg. Similarly, the handling times at the various logistic nodes differ based on the time of the day. At the beginning of every simulation run, the transport orders for the trucking company are checked and each truck is assigned one order. When the first order is completed, a next order is chosen for the truck. When a truck arrives too late at its destination, the order is cancelled and a new order is assigned.

4 Experimental Design

The parameters for the simulation experiments were determined in interviews with different stakeholders in the drayage network in the port of Hamburg. In addition, relevant scientific publications were analyzed. In the simulation experiments, a medium-sized trucking company with 25 trucks and 375 possible orders per day is considered. 50 % of the transports are executed between container terminals and 8.3 % each be- tween container terminals and empty container depots, packing stations and the remaining logistics nodes, vice versa. Transports only between empty container depots, freight stations and other logistics nodes are not considered. Furthermore, the working times of all logistics nodes, except container terminals, which always have three shifts, are limited. Therefore, they restrict the productivity of drayage transports as well as the efficiency of TAS at container terminals. As for TAS' characteristics, the capacity of the time windows in the different shifts (based on current demand, lightly smoothed and heavily smoothed) is varied. In the first shift there are either 40, 60 or 80 slots in total. In the second shift there are 120, 100 or 80 and in the third are always 80 slots. Furthermore, the utilization of the time windows by other trucking companies (80 %, 85 % and 90 %) is varied for the peak times between 9 am and 6 pm. The peak time has been set based on data provided by logistics nodes in the port of Hamburg. The overview of all experiments and their parameters is shown in Table 1.

	Utilization		
Capacity	**80 %**	**85 %**	**90 %**
Realistic	Exp. 1	Exp. 4	Exp. 7
Lightly smoothed	Exp. 2	Exp. 5	Exp. 8
Heavily smoothed	Exp. 3	Exp. 6	Exp. 9

Table 1: Plan of experiments.

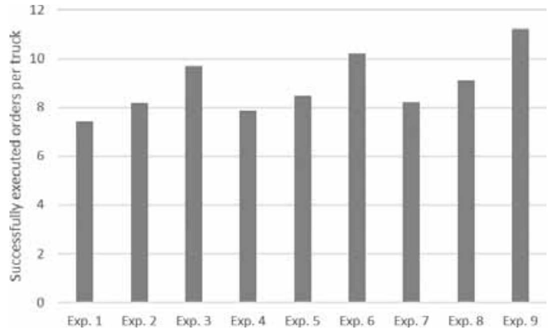


Figure 1: Successfully executed orders per truck and day.

5 Simulation Results

The results show a high impact of the two selected TAS' characteristics on the operative stakeholders. In Figure 1, the amount of successfully executed orders is shown. It is surprising to see that the heavily smoothed TAS capacity leads to a better solution as well as the high capacity utilization by other trucking companies. This effect is very likely caused by the structure of transport orders generated. As a high amount of transports is executed between container terminals and only a lesser amount between container terminals and other logistics nodes, vice versa, the trucks are able to pick up and deliver containers 24/7. In the simulation model, the time windows at peak times have a higher priority and are therefore, chosen first. If no time windows in peak hours are left, the time windows in off-peak hours are booked. This leads to shorter waiting times at the terminals due to the lower assumed handling times as well as to less congestion in the port and thereby, to lower trans- port times. Furthermore, transports in the peak times tend to be more risky due to a higher variance in trans- port and handling times. Therefore, more transports in peak hours need to be cancelled due to predicted late arrivals at their destination. This reduces the productivity of the trucking companies considerably.

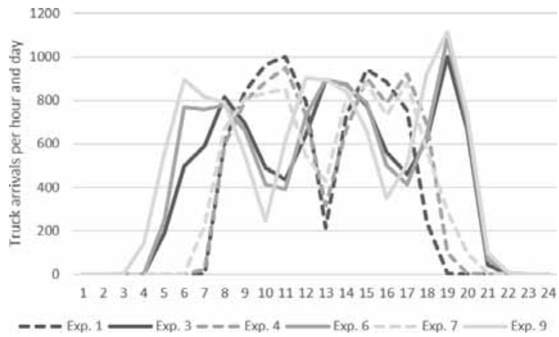


Figure 2: Truck arrival times at all types of logistics nodes for Exp. 1,3,4,6,7 and 9.

A similar effect can be seen in Figure 2. There, the arrivals at logistics nodes are spread out in a wider time range for Exp. 3, 6 and 9 compared to Exp. 1, 4 and 7. Considering the lower transport and handling times at the off-peak hours, this leads to a higher number of executed transports and therefore, to a higher productivity for the trucking companies. The transports executed in the off-peak hours are mainly inter-terminal transports, as the other logistics nodes tend to have limited working hours. This fact can also be seen in Figure 3, where the truck arrival times at container terminals and other logistics nodes are shown exemplarily for Exp. 1 and 9. It is evident that the inter-terminal transports in Exp. 9 are further shifted to off-peak hours. A high percent- age of the transports in peak hours is executed between container terminals and other logistics nodes.

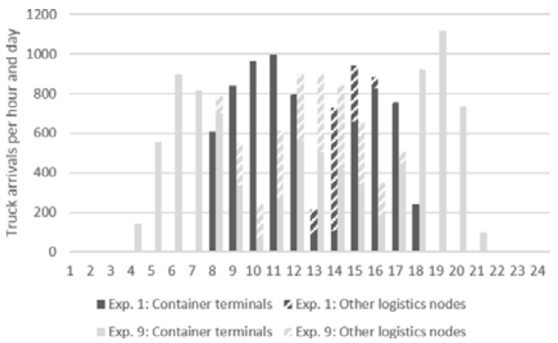


Figure 3: Detailed truck arrival times for Exp. 1 and 9.

6 Conclusion and Outlook

In conclusion, a TAS offers chances for trucking companies as well as for logistics nodes. Especially, if a trucking company executes many inter-terminal transports it is flexible enough to adapt to a TAS and seize the opportunities. It is expected that the results would change considerably if a higher amount of non-inter-terminal transports is assumed. In this case, the restrictions imposed by TAS and limited opening hours of other logistics nodes would very likely reduce trucking companies' productivity when the available capacity is limited. In future, more TAS' characteristics should be analyzed on their impact on the different stakeholders in drayage networks. Furthermore, a broader variance in booking strategies for the trucking companies as well as operations at logistics nodes should be considered.

References

[1] United Nations Conference on Trade and Development. *Review of maritime transport 2017*. New York, Geneva: United Nations; 2017

[2] Hartmann S. Generating scenarios for simulation and optimization of container terminal logistics. *OR Spectrum*. 2004; 26, 171–192.

[3] Huynh N, Harder F, Smith D, Sharif O, Pham Q. Truck Delays at Seaports. *Transportation Research Record: Journal of the Transportation Research Board*. 2011; 2222, 54–62.

[4] Harrison R, Hutson N, West J, Wilke J. Characteristics of Drayage Operations at the Port of Houston, Texas. *Transportation Research Record: Journal of the Transportation Research Board*. 2007;2033: 31–37.

[5] Shiri S, Huynh N. Optimization of drayage operations with time-window constraints. *International Journal of Production Economics*. 2016; 176: 7–20.

[6] Giuliano G, O'Brien T. Reducing port-related truck emissions. The terminal gate appointment system at the Ports of Los Angeles and Long Beach. *Transportation Research Part D: Transport and Environment*. 2007; 12: 460–473.

[7] Bentolila DJ, Ziedenveber RK, Hayuth Y,Notteboom T. Off-peak truck deliveries at container terminals: the "Good Night" program in Israel. *Maritime Business Review*. 2016; 1/1(1):2–20.

[8] Davies P, Kieran M.E. Port Congestion and Drayage Efficiency. *METRANS International Urban Freight Conference*. 2015 Oct; Long Beach, USA.

[9] Huynh N, Smith D, Harder F. Truck Appointment Systems. *Transportation Research Record: Journal of the Transportation Research Board*. 2016; 2548: 1–9.

[10] Chen G, Govindan K, Yang Z-Z, Choi T-M, Jiang L. Terminal appointment system design by non-stationary M(t)/Ek/c(t) queueing model and genetic algorithm. *International Journal of Production Economics*. 2013; 146: 694–703.

[11] Zhao W, Goodchild AV. The impact of truck arrival information on container terminal rehandling. *Transportation Research Part E: Logistics and Transportation Review*. 2010; 46: 327–343.

[12] Heilig L, Lalla-Ruiz E, Voß S. port-IO. An integrative mobile cloud platform for real-time inter-terminal truck routing optimization. *Flex Serv Manuf J*. 2017; 29: 504–534.

[13] Namboothiri R, Erera AL. Planning local container drayage operations given a port access appointment system. *Transportation Research Part E: Logistics and Transportation Review*. 2008; 44: 185–202.

[14] Lange A-K, Schwientek A, Jahn C. Reducing truck congestion at ports – classification and trends. In: Jahn C, Kersten W, Ringle C M, editors. Digitalization in maritime and sustainable logistics. City logistics, port logistics and sustainable supply chain management in the digital age. *Hamburg International Conference of Logistics*. 2017 Sep; Hamburg, Germany:epubli. 37–58.

Agent-based Simulation of Job Shop Production

Tao Zhang*, Shufang Xie, Oliver Rose

Fakultät für Informatik, Universität der Bundeswehr München,
Werner-Heisenberg Weg 39, D-85577 Neubiberg, Germany; *tao.zhang@unibw.de

SNE 29(3), 2019, 141 - 148, DOI: 10.11128/sne.29.tn.10487
Received: June 15, 2019 (Selected ASIM SST Hamburg 2018 Postconf. Publ.), Accepted: July 22, 2019
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. The most important design principles of Industry 4.0 are decentralization and intelligence. The entities, like resources and materials, can make decisions on their own by means of cyber-physical systems. For research purposes, we unify cyber entities and physical entities and build an agent-based simulation model. The agents learn knowledges offline during simulation runs and become smarter and smarter. The model will finally connect to the physical systems and carry out online decision-making. The study is the first stage of the whole project. A framework for the agent-based simulation is developed and an agent-based model of the job shop production including release agents, machine group agents, and job agents are built.

Introduction

Industry 4.0 [1] is currently one of the most frequently discussed topics among practitioners and academics. Taking advantages of cyber-physical systems [2], the internet of things [3] and the internet of services, it enables faster, more flexible, and more efficient processes to produce higher-quality goods at reduced costs in an environmental-friendly and resource-conserving way. This in turn will increase manufacturing productivity, shift economics, foster industrial growth, and modify the profile of the workforce – as ultimately change the competitiveness of companies and regions. Most of studies and industry cases about Industry 4.0 still stay at a very basic level, such as studies and cases about sensors, communications, and automations. These are certainly foundations of Industry 4.0. However, once these foundation works are done, there will be a very urgent requirement for more effective decision-making methods on the platform of Industry 4.0.

As we all know, the most important design principles of Industry 4.0 are decentralization and intelligence [4]. The cyber-physical systems can make decisions on their own. To make the cyber-physical systems intelligent, the systems must have abilities to learn knowledge. Online learning and offline learning are two common learning techniques. For the research purposes, the online learning is unpractical. In the study, we will create an agent-based simulation model (cyber system) and the agents (cyber entities) will learn to make decisions from the simulation. At last, the model will connect to the physical systems and form the cyber-physical systems.

The study is an extension of our previous works [5, 6] in which we realized that the agent-based simulation (ABS) is less efficient when being used into a non-real-time system even though it can be speeded up by giving a timescale. So, to speed up the ABS, we introduced a process-interaction worldview (PIW) originated in the discrete event simulation to the ABS.

In this study, we are going to design a framework for the ABS with the PIW and build a model of the job shop production based on the framework. In the model, each agent is related to one physical entity. The agents can make decisions by either some decision rules or their knowledge learnt from some other independent simulation-based approaches that we have proposed [7-9].

1 Agent-based Simulation with PIW

The ABS&PIW approach [5, 6] was proposed on the basis of the agent-based model(ABM). We provided a four-tuple $SIM = (I, TM, ABM, O)$ with elements inputs (I), time manager (TM), ABM, and outputs (O) to describe the approach strictly. The procedure of the approach is presented mathematically in which the simulation clock advances in a sequence of activation points and all concurrent activations are activated at a time, and associated agents respond in parallel.

To reuse the code and make the development of the ABS with the PIW easy, in this section we will give a general idea that how to develop a framework for the

ABS&PIW. We will design individual agent and time manager, and draw a static structure of the framework.

1.1 Individual Agent

The structure of the agent is shown in Figure 1. The agent is composed of attributes, an initialization method, a behavior controller, and a message handler.

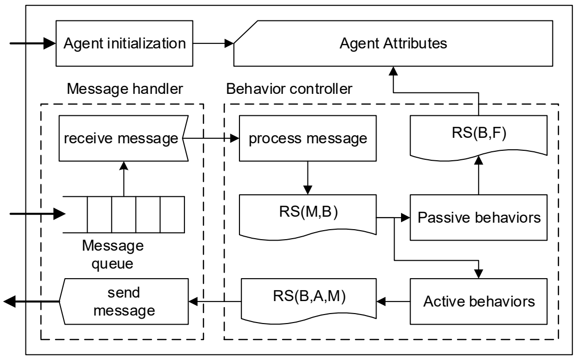


Figure 1: Structure of individual agent.

Agent Initialization

Agent initialization, which is applied to set the values of attributes when an agent is created, is the only possibility to change the state of an agent directly by the external environment and enables the simulation to start in any state of the agent. There are also many other common methods (e.g., reset, start, and stop) for controlling agents, but they are not visible to the environment and called only by agents themselves.

Behavior Controller

The behavior controller decides which behavior to be executed when receiving a message. The decision is made depending on the relationships RS. Here we just summarize them in three types of relationships: RS (M, B) is the relationship between messages (in) and behaviors; RS (B, F) maps attributes to behaviors; RS (B, A, M) denotes the relationship between behaviors and messages (out). The behavior controller can also control behaviors, such as adding behavior, removing behavior, and changing behavior's state according to the environment.

Message Handler

Through the message handler, the agent communicates with other agents. A message handler includes one message queue and two methods: "send" and "receive." Messages from other agents will be stored in the message queue and received by "receive" method. Similarly, the agent can send the messages to other agents by using the method "send." These two methods are behaviors of the agent.

1.2 Time Manager

In order to keep consistent with the agent-based model, the time manager is also developed as an agent to be in charge of advancing time, activating agents, and managing activation points. The time manager extends the class of agents, and Figure 2 shows its structure.

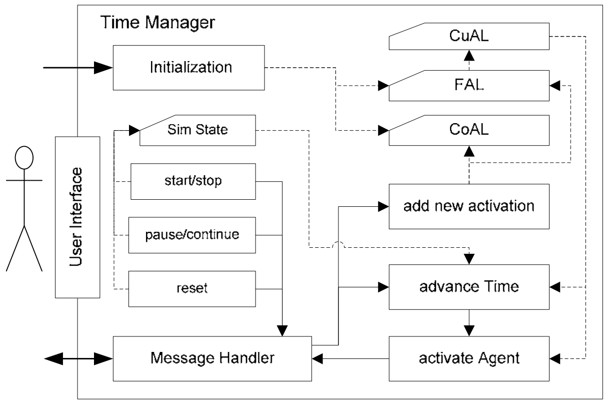


Figure 2: Structure of the time manager.

Behavior of the Time Manager

There are three main behavior types: advance simulation clock, receive activation point and activate agent. A user interface is provided to control the simulation. Before the simulation runs, at least one activation point needs to be given in advance. Activation points are conveyed in the form of messages between the time manager and other agents. After the activation, the time manager is blocked until a new activation point is received or the ABM notifies it to advance the simulation clock when the model state is ready.

Activation Point Lists in the Time Manager

Three activation point lists are created in the time manager: conditional activation list (CoAL), future activation list (FAL) and current activation list (CuAL).

The time manager puts new received activation points into the appropriate list. The earliest activation points are moved from the future list to the current list every time the simulation clock advances. After activation, the current list is cleared. Conditional activation points are tried again and again and removed from the list when the conditions are met.

1.3 Agent-based Model

The agent-based model includes the agent environment, the agent manager, and a set of agents (shown in Figure 3).

The agent environment is a medium for communication among the agents. The time manager and the agents send or receive activation points in the environment too. The agent manager is in charge of all agents and provides the model state to the time manager when the simulation is running. A new agent needs to register with the manager. The agents report their physical states and updated flags when they become blocked. The time manager also needs to register. In the model, each agent has a unique name which is used to specify the target agent in the communication.

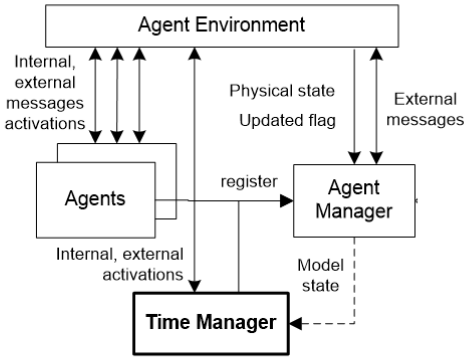


Figure 3: Structure of the Agent-based Model.

2 Agents in Job Shop Production

Agent-based modeling enables us to model the job shops more realistically and systematically comparing to other modeling methods, like discrete event simulation modeling, Petri nets, and so on. We know that an agent-based model is composed of agents and their relationships. Thus, we should determine who the agents are in the job shop production first. Obviously, material plays a leading role in the job shops. However, materials are usually transformed to other types of materials in shops, and their lifecycle is very short. Paying attention to them makes no sense.

We usually focus on jobs which organize all materials that one product needs in a serial of sequential operations. Besides, machines and transporters also make up a large proportion of the job shops. Because we assume that the transportation capacity is finite, the transporters are out of scope. The transportation in our study is treated as a delay. We only consider the machines here. As a type of flow, the material flow must have one or more sources and let jobs derive from it. The sources are usually job pools. Each type of job, i.e., product, is connected to a job pool.

There is also a valve controlling the flow from each pool. In the job shops, this valve is a job release procedure. The release procedure is what we will concentrate rather than the job pools. To date on the basis of analysis above, we list three entities concerned in the job shops: job, machine group, and release procedure. We will model these three entities as agents in the agentbased model: release agent, job agent, and machine agent. In this section, we will describe how to create them on the basis of the framework, including their attribute and behavior definitions, communication and cooperation design, and simulation-related delay and activation design.

2.1 Release Agent

In a release agent, there are one piece of product data, multiple release policies, and one buffer. The release agent creates the job agents who are given in the product data based on a release policy. The buffer is located behind the release agent (see Figure 4). If the corresponding buffer is full in the first operations, the released job cannot start the operations. It will be blocked and stays in the release buffer until the buffer of the first operations has free space. If the buffer of the release agent is full, the release agent stops releasing until the buffer is not fully occupied.

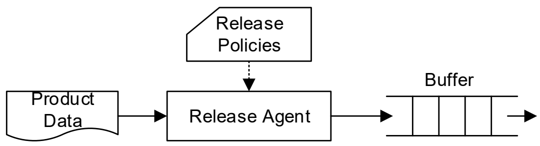


Figure 4: The release agent with one buffer.

Product Data in the Release Agent

At the beginning of the simulation, the simulator reads product and process data from files and generates each product a release agent.

Each product has a unique name and five data elements: a production probability, process, priority, the release interarrival time, and the target work-in-process (WIP) level. The process element is the name of the process flow which is defined in the process file. Once the job agents are created, they will obtain their process flows from the product. The priority specifies the urgency of products. The value of the priority is from 0 to 1 and is used by some dispatch rules. The elements of interarrival time and WIP level are used by the release policies.

Release Policies of the Release Agent

The release policy is the criterion which decides when to release jobs. The release policies are a very common way to solve the job release problem. For now, the release agent has three very common types of release policies: constant interarrival time (CONINT), constant WIP level (CONWIP), and avoiding starvation (AS). CONINT releases the job in the same interval. To the policy of CONWIP, one interval time is still needed to be designated before reaching the target WIP. After reaching the target WIP, the policy of CONWIP takes effect. If the WIP level is less than the target WIP level, a new job is released. In the AS policy, a target buffer size is set for a bottleneck machine group. When the buffer size of the bottleneck is more than the target value, the releasing stops. When the buffer size is less than the target, new jobs are released. The quantity of the released jobs is the difference between the buffer size and the target value.

Communication with the Time Manager and Other Agents

If the release policy is CONINT, "releasing job" is an activation point which will be sent to the time manager and put in the activation list. When reaching the activation time, the time manager will send an activation message to the release agent. Receiving the activation message from the time manager, the release agent will be activated and start to release jobs. In the case of CONWIP, the finished job will send a message to the related release agent which will release a job immediately after receiving the message. For the AS policy, the machine agent of the bottleneck will request the release agent to stop releasing or ask the release agent to release jobs. If a released job is refused by the machine agent in the first operation due to the fullness of its buffer, the job sends blocked message to the release agent, and the release agent puts it into the buffer. When accepted, the job sends unblocked message to the release agent. The job will be removed from the buffer and moved to the first operation.

2.2 Job Agent

The job agent is a temporary entity. After released, it will be processed on many machines in the order of its process flow which depends on the target product, and after finished it will be destroyed. The job agent has a unique name in the model and four logical states: transporting, waiting, processing, and blocking.

Behaviors and Life cycle of the Job Agent

A job agent has five behaviors: to request resources, to be transported, to enter the buffer, to wait, to be processed, and to be blocked. The lifecycle of a job agent is shown in Figure 5. After release, the job requests the next operations. If accepted, it begins transporting and then enters the buffer to wait. If refused, it is blocked on the current machine. After receiving the start message from the machine group, the job starts. The job finishes when receiving the end message from the machine group and then requests the next operation. If it is the last operation, the job completes.

2.3 Delays and Activations in the Job Agent

There are four types of delays related to the logical state: transporting, blocking, waiting and processing. Blocking delay means that a finished job cannot move to the next operation due to the fullness of the related buffer and will continue to stay on the current machine. This is a conditional delay, and it will be activated by the time manager every time the simulation time advances. When a job begins to be transported or is blocked, associated activation points (transporting end, blocking end) will be sent to the time manager. After that, the job agent will wait for activations from the time manager and the delay will end when it receives the activation messages. The waiting delay and the processing delay end when the job receives the related message from a machine.

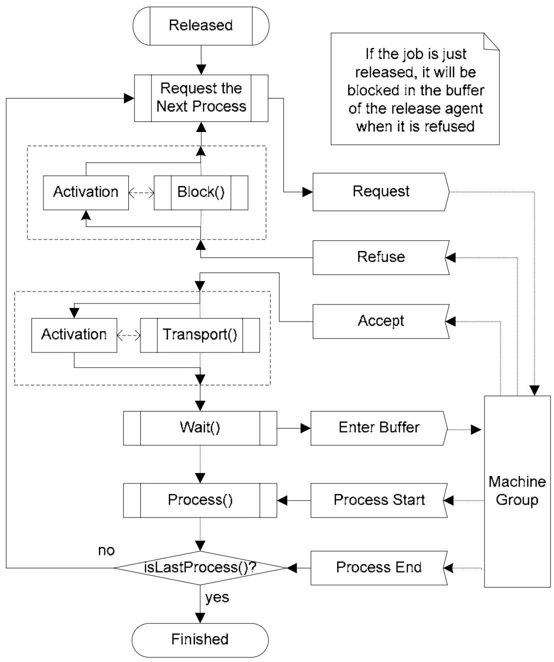


Figure 5: Lifecycle of a job agent.

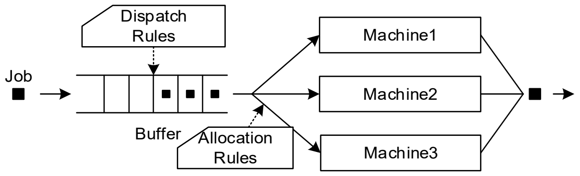


Figure 6: Structure of the machine group.

2.4 Machine Group Agent

The machine group agent is a permanent and active entity. It includes one buffer and several machines (see Figure 6). The machines have the same function and share the buffer.

There are five logical states of the machines: idle, busy, setup, breakdown, and maintenance. The machines may be single processing machines or batch processing machines. When a job arrives, and cannot be processed at once, the job joins the buffer. The buffer has a finite capacity, and it dispatches the waiting job to the idle machine according to a dispatch rule. There is no buffer behind the machine group. When a job finishes, the ther be transported to the next machine group or be blocked on the current machine.

Behaviors of Machine Group Agent

The machine group agent has two behaviors: to respond to the request from the job agent and to select the best machine to process the job. The buffer has two behaviors: to store the job and to dispatch the stored jobs to the machines in a certain batch and the order of the given priority. The machines have five behaviors: to request the jobs from the buffer, to setup, to process, to interrupt, and to recover.

When a job enters a buffer or a machine just finishes one job, the machine group will decide on the start of a new process. If the buffer is not empty (in case of batch processing, a batch must be ready) while the machine group has an idle machine, the idle machine will start processing and inform related job agents. If there is more than one idle machine, one machine will be selected according to an allocation rule. If the setup is needed, the machines will start the process after the setup. Figure 7 shows the behavior flow of the machine group agents.

Dispatch and Allocation Rules in Machine Group Agent

A dispatch rule is a criterion for determining the jobs' process priority in the buffer. An allocation rule is for selecting one machine from the idle machines. The dispatch and allocation rules are very common ways to solve the sequencing and routing problems.

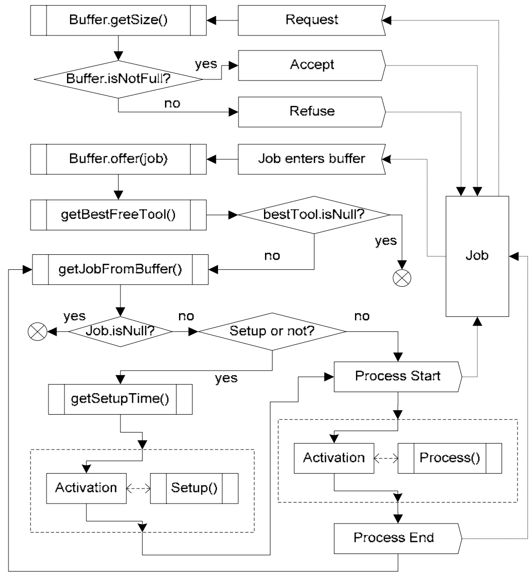


Figure 7: Behavior flow of a machine group agent.

A buffer may include a set of dispatch rules. When a job joins in the buffer, all messages associated with dispatch rules are sent to the buffer. Based on these messages and under a given dispatch rule, the buffer determines the priority of each job and queues them in that sequence. Once a job joins the buffer, the priority is updated. Currently, 17 types of common dispatch rules, such as FIFO (First In First Out), EDD (Earliest Due Date), CR (Critical Ratio), etc., have been preset in the buffer. The allocation rules are used by the group agent.

Once a job comes out from the buffer, the group agent will collect information of all machines and select one to process the job.

Delays and Activations in Machine Group Agent

There are four types of delays related to the logical state in the machine group agents: setup, processing, breakdown and preventive maintenance. When the setup delay occurs, the machine group agent sends an activation message (setup end) to the time manager and informs related jobs. The jobs will wait and cannot be processed by other machines. When the setup delay ends, the machine group agent sends an activation (process end) message to the time manager. Meanwhile, it informs the jobs and starts processing. Activated by the time manager, the machine finishes processing and informs the jobs.

2.5 Communication among the Agents

We summarize the communication in Figure 8. It includes the delays and activations conveying between the time manager and the agents as well as the communication among the agents.

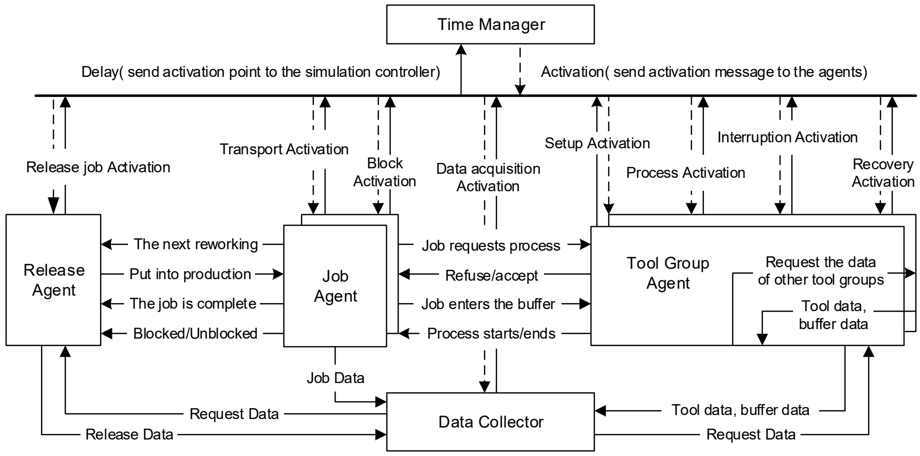


Figure 8: Communications among the agents and the time manager.

3 Agent-based Simulation of the Job Shop Production

3.1 Overview of the Simulation

The agent-based simulation of the job shop production consists of a simulator, production data, and performance measures (see Figure 9). The production data are inputs, and the performance measures are outputs.

Production Data

We classify the production data into three parts: machine group data, product data, and process data, and then adopt XML to store the data. Thus, three XML files are created to store the data. The machine group file includes data about all machine groups, e.g., the number of machines, buffer size, dispatching rule, etc. The process file consists of the process flows of all products. Each processing flows has many operations. The product file consists of buffer size, waiting time (by machine group), blocking data about all products which will be produced. Each product has a name of the process flow which is defined in the process file. The process file links the product file and the machine group file together.

Simulator for the Job Shop Production

The agent-based model (including release agents, machine group agents, and job agents), data collector and time manager make up the simulator for the job shop production. The time manager creates the release agents and the machine group agents according to the machine group data at the beginning of the simulation, and it is responsible for advancing the simulation time and handling the simulation control (e.g., start, stop, pause, etc.).

The data collector is responsible for collecting simulation data and computing the performance measures. It can collect all simulation data in detail, as well as part of sample data.

Performance Measures

The performance measures include WIP level, cycle time, time (by machine group), and machine utilization information (e.g., idle time, processing time, breakdown time and setup time). All data can be shown in graphs and tables. These measures can be used to improve the job shop production and can be provided for the optimization or control algorithms to achieve the goal of the optimization and control.

3.2 Static Structure of the Simulator

The static structure of the simulator (see Figure 10) has three layers: framework layer, agent layer, and production characteristics layer.

The framework layer provides the agent base class and the time manager. The agent layer contains the agents which are abstracted from the job shops, and these agents extend the base class of the agent in the framework. A machine group agent has multiple machines and one buffer. Each release agent can release one kind of product.

The data collector, which is also an agent and in charge of data collection from machines, buffers, jobs, and release agents, is contained in the agent layer. In the production characteristics layer, many characteristics are considered, such as reentrant flows, rework, setups, batch processing, breakdowns, and preventive maintenance. Dispatch rules and release policies are part of this layer.

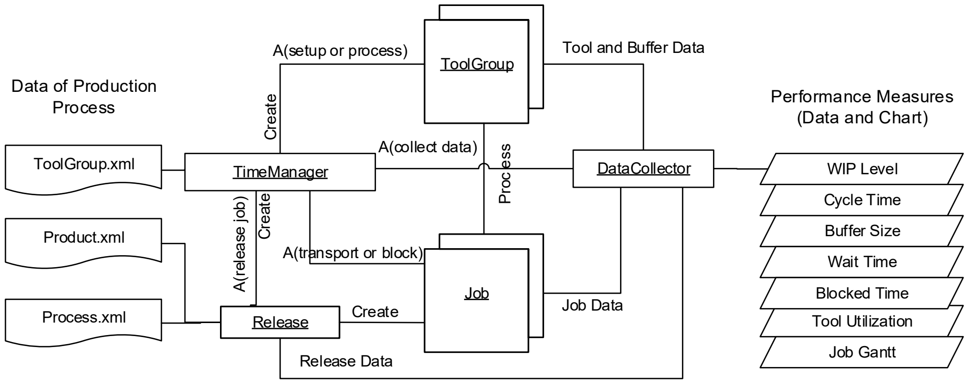


Figure 9: Agent-based simulation of the job shop production.

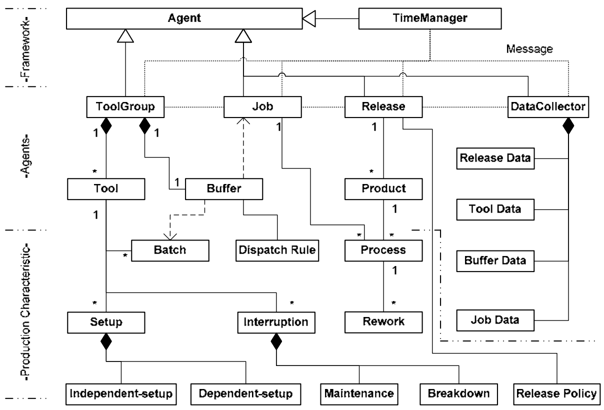


Figure 10: Static structure of simulator.

4 Applications

We create a simulation model of an example job shop. The job shop contains five machines and produces two products (Pa and Pb) with two process flows and two different throughputs. The throughputs of product Pa and Pb are 110 and 197 units per week. There are no batch processing machines. The machines need sequence-dependent setups. The interval between two breakdowns on the machines is subject to the exponential distribution and the repairing time follows an exponential distribution too.

The simulator is used to solve real-time sequencing problems in the job shop in which we should decide which job should be processed first once a machine becomes available. The objective is to minimize the cycle time. The sequencing problems are sequential decision-making problems. The decision-making is the selection of the best job from possible jobs. In other words, it is a priority calculation problem. For each job, a priority value is computed. The job with the highest priority is selected.

We proposed three simulation-based approaches to calculate the priority values of jobs, including the simulation try-then-decide method (STTD) [8], the intelligent method based on the simulation try-then-decide method (INT1) [7], and the intelligent method based on Markov decision process (INT2) [9].

The STTD method is a pure simulation approach. It uses the alternative simulations to predict the future after a job is taken and select jobs according to the future information from the simulation. The most important innovation is the usage of the base-rule in the alternative simulations. The base-rule avoids the exponential explosion of the number of the alternative simulations. To evaluate the STTD method, we replace the system with an environment simulation. The decisions we made are executed in the environment simulation. The results from the environment simulation can be used to evaluate the STTD method.

The INT1 method combines the experiences & data approach with the simulation approach. A data-driven model is introduced to calculate the priority values for jobs. It is built on the data from the simulation with the STTD method. So, it manages to learn the knowledge of the STTD method. Two types of factor influence the priority value of the job: global factors and local factors. The state of the job shops is divided into several patterns by clustering the global data. In each pattern, the priority value is only up to the local factors. The relationship between the priority and the local factors is mapped in the neural networks. For each pattern, one neural network is created. In the decision maker, the centroids of the patterns make up a pattern pool, and the neural networks make up a network pool. While making the decision, the decision maker determines the pattern of the current state according to the pattern pool and selects one corresponding neural network from the network pool.

The neural network will calculate the priority for each job according to the local factors.

The INT2 method combines the experiences & data approach, mathematical approach, and simulation approach. It models the sequencing as Markov decision process with multiple decision makers. We defined the five-tuple for the problems, including decision points, state space, action sets, transition procedure, and a reward function.

The data-driven model is still used to map the value of the action to the state-action pair. The simulation-based batch-mode Q-learning algorithm explores the state space by the simulation. Each simulation run is one iteration. The data-driven model is updated and improved gradually after each iteration.

The three methods compare with each other as well as other two decision rules, First In First Out (FIFO) and Shortest Processing Time (SPT). The experiment results are shown in Table 1. The results show that the STTD and INT1 methods always outperform the rules. The STTD method performs best but consumes much time. Contrarily, the INT1 method takes less time while the performance is just a little bit worse than the STTD. Thus, the STTD is more suitable for offline applications, and the INT1 can be used in the real-time control. Unfortunately, the INT2 method performs unsteadily. It will be further studied in the future.

5 Conclusions

On the basis of the previous works, a framework for the ABS is designed and an agent-based model of the job shop production including release agents, machine group agents, and job agents are built. The behaviors and interactions among the agents are explicit defined. A large variety of decision rules are preset in the model as well. The agents can make decisions simply according to the rules. The agent-based model, the data collector, and the time manager make up the simulator for the job shop production. Applying the simulator to an example job shop, we solved the real-time sequencing problem by three simulation-based approaches we have proposed. The application and experiment results indicate that the overall idea of simulation-based learning and decision-making is feasible in the job shops.

References

[1] Lasi, H., et al., *Industry 4.0.* Business & Information Systems Engineering, 2014. **6**(4): p. 239-242.

[2] Jazdi, N. *Cyber physical systems in the context of Industry 4.0*. in *2014 IEEE International Conference on Automation, Quality and Testing, Robotics*. 2014.

[3] Wortmann, F. and K. Flüchter, *Internet of Things.* Business & Information Systems Engineering, 2015. **57**(3): p. 221-224.

[4] Hermann, M., T. Pentek, and B. Otto. *Design Principles for Industrie 4.0 Scenarios*. in *2016 49th Hawaii International Conference on System Sciences (HICSS).* 2016.

[5] Xie, S., T. Zhang, and O. Rose, *Agent-based Simulation with Process-interaction Worldview*, in *ASIM 2018 24. Symposium Simulationstechnik*. 2018: Hamburg.

[6] Zhang, T. and O. Rose, *A framework for agent-oriented parallel simulation of discrete event systems*, in *Proceedings of the Winter Simulation Conference*. 2012, Winter Simulation Conference: Berlin, Germany. p. 1-2.

[7] Zhang, T. and O. Rose. *Intelligent dispatching in dynamic stochastic job shops*. in *2013 Winter Simulations Conference (WSC)*. 2013.

[8] Zhang, T. and O. Rose, *Simulation-based Dispatching in Job Shops*, in *ASIM 2014 22. Symposium Simulationstechnik*. 2014: Berlin.

[9] Zhang, T., S. Xie, and O. Rose. *Real-time job shop scheduling based on simulation and Markov decision processes*. in *2017 Winter Simulation Conference (WSC)*. 2017.

ARGESIM Benchmark C11 'SCARA Robot': Comparison of Basic Implementations in EXCEL and MATLAB

Olga Rekova[1,2], Nicole Pelzmann[1,2], Philipp Mandl[1,2], Maximilian Hoffmann[1,2],

Horst Ecker[2*], Andreas Körner[1], Martin Bicher[1,3], Felix Breitenecker[1]

[1]Mathematical Modelling and Simulation Group, Inst. of Analysis and Scientific Computing
[2]Inst. of Mechanics and Mechatronics, [3]Inst. of Information Systems Engineering
TU Wien, Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria; **horst.ecker@tuwien.ac.at*

SNE 29(3), 2019, 149 - 158, DOI: 10.11128/sne.29.bne11.10488
Received: November 10, 2018; Revised: March 13, 2019;
Revised: July 25, 2019; Accepted:July 30, 2019
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. This SCARA robot benchmark study documents an EXCEL implementation and compares it with a MATLAB implementation on basis of Euler and Heun ODE solvers with integrated event handling. Aim is to check whether a spreadsheet tool like EXCEL can be used as simulator for continuous models as given for the SCARA robot. The benchmark study is organized in four parts. The first part describes the proper transformation and preparation of the implicit model description into an explicit state space with integrated control and state restrictions for use with explicit Euler and Heun solver. The second part documents the model implementations in EXCEL and in MATLAB for simulation of point-to-point movement. The third part concentrates on model extension for obstacle avoidance and proper implementation in EXCEL and in MATLAB. The fourth part compares the simulation results on a numerical basis and discusses advantages and disadvantages of the implementations. An appendix shows snapshots from the EXCEL implementations.

Introduction

ARGESIM Benchmark C11 'SCARA Robot' is based on a mechanical model for a three-axis SCARA robot (**S**elective **C**ompliance **A**ssembly **R**obot **A**rm).

The three degrees of freedom are constituted by two vertical revolute joints (angles q_1, q_2) and one vertical prismatic joint (distance q_3) as shown in Figure 1.

Such a system can be fully described by an implicit second-order system of differential equations:

$$M\ddot{\vec{q}} = \vec{b} \tag{1}$$

Here $\ddot{\vec{q}} = (\ddot{q}_1, \ddot{q}_2, \ddot{q}_3)^T$ represents the second derivative of the joint vector, and M is the mass matrix, which has a block-diagonal form and can be inverted symbolically:

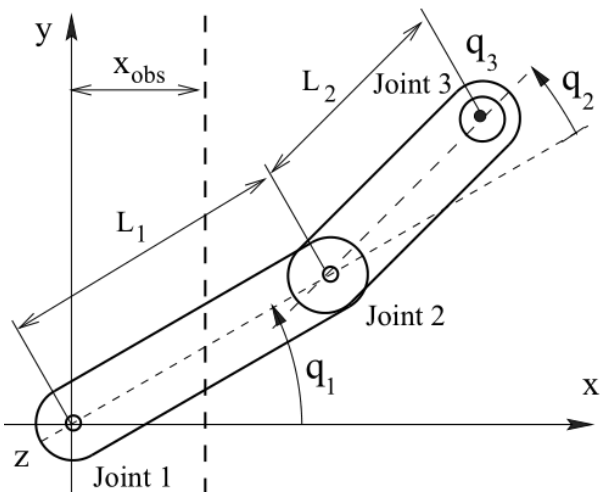


Figure 1: Three-axis SCARA robot, three degrees of freedom q_1 (rotational), q_2 (rotational), q_3 (translational). [1].

$$M = \begin{bmatrix} ma_{11} & ma_{12} & 0 \\ ma_{21} & ma_{22} & 0 \\ 0 & 0 & ma_{33} \end{bmatrix} \quad (2)$$

The components of the mass matrix are given in the definition of the benchmark [1], but can also easily be derived using Lagrangian mechanics (neglecting the rotational kinetic energy of the load m_3L and of the motor for the vertical axis):

$$\begin{aligned} ma_{11} &= \Theta_1 + 2\Theta_2 cos(q_2) + \Theta_3 \\ ma_{12} &= \Theta_2 cos(q_2) + \Theta_3 \\ ma_{21} &= ma_{12} \\ ma_{22} &= \Theta_3 \\ ma_{33} &= m_{3L} + \Theta_{3mot} u_3^2 \end{aligned}$$

Here Θ_i are the moments of inertia. These moments are calculated based on the assumption that the two physical links are two rods of mass m_1 and m_2 with homogeneous mass distribution along their length L_1 and L_2.

The right-hand side $\vec{b} = (b_1, b_2, b_3)^T$ is made up of the following equations, whereby T_1 and T_2 are the joint torques and T_3 is the joint force - inputs for the uncontrolled systems:

$$\begin{aligned} b_1 &= T_1 + \Theta_2(2\dot{q}_1\dot{q}_2 + \dot{q_2}^2) sin(q_2) \\ b_2 &= T_2 - \Theta_2 \dot{q_1}^2 sin(q_2) \\ b_3 &= T_3 - m_{3L} g \end{aligned}$$

For operation, servo motors for each axis drive the robot following a specific control scheme (joint torques and joint force are proportional to the current of the respective motor). The electrical relationship of the armature of a robot servo motor is given by a first order differential equation for the current I_i of the servo motors, whereby the current I_i must be limited to I_{ai}:

$$\dot{I}_i = g_{I,i} = \frac{U_{ai} - k_{T_i} u_i \dot{q}_i - R_{ai} I_{ai}}{L_{ai}}, \; i = 1,2,3$$

$$I_{ai} = [-I_i^{max} \leq I_1 \leq I_i^{max}], \; i = 1,2,3 \quad (3)$$

Here k_{T_i}, u_i, R_{ai}, and L_{ai} are parameters, the control voltages U_i and U_{ai} resp. result from PD control for point-to-point movement with target joint position vector $\hat{\vec{q}} = (\hat{q}_1, \hat{q}_2, \hat{q}_3)^T$:

$$U_i = P_i(\hat{q}_1 - q_i) - D_i \dot{q}_1], i = 1,2,3$$

$$U_{ai} = [-U_i^{max} \leq U_1 \leq U_i^{max}], i = 1,2,3 \quad (4)$$

1 Explicit State Space Model

Challenge of this SCARA robot benchmark report was a proper implementation in the spreadsheet tool EXCEL and the comparison with a (basic) MATLAB implementation. EXCEL is no simulator, it does not provide ODE solvers or other simulation tools. But simple ODE solvers can be easily implemented by means of recursive cell update in columns. Consequently, solver choice was explicit Euler ODE solver and explicit Heun ODE solver, for an explicit state space

$$\dot{\vec{x}} = \vec{f}(\vec{x}) \quad (5)$$

given on a grid $t_0, t_1, ..., t_n$, $\vec{x}_i = \vec{x}(t_i)$ with constant stepsize $h = t_{i+1} - t_i$ by

$$\begin{aligned} \vec{x}_{i+1}^E &= \vec{x}_i + h \cdot \vec{f}(\vec{x}_i) \quad (6) \\ \vec{x}_{i+1}^H &= \vec{x}_i + \frac{h}{2} \cdot (\vec{f}(\vec{x}_i) + \vec{f}(\vec{x}_i + h \cdot \vec{f}(\vec{x}_i))) \\ &= \vec{x}_i + \frac{h}{2} \cdot (\vec{f}(\vec{x}_i) + \vec{f}(\vec{x}_{i+1}^E)) \quad (7) \end{aligned}$$

To formulate the robot with servo motors and control in explicit state space form (5), first the second derivatives in (1) were replaced by three additional states, and the currents for the servo motors in (3) were also integrated in the state space, resulting in a 9 by 9 implicit system:

$$A(\vec{x}) \cdot \dot{\vec{x}} = \vec{g}(\vec{x}) \quad (8)$$

$$A = \begin{bmatrix} ma_{11} & ma_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ ma_{21} & ma_{22} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & ma_{33} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{aligned} \dot{\vec{x}} &= (\ddot{q}_1, \ddot{q}_2, \ddot{q}_3, \dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{I}_1, \dot{I}_2, \dot{I}_3)^T \\ \vec{x} &= (\dot{q}_1, \dot{q}_2, \dot{q}_3, q_1, q_2, q_3, I_1, I_2, I_3)^T \\ \vec{g}(\vec{x}) &= (b_1, b_2, b_3, \dot{q}_1, \dot{q}_2, \dot{q}_3, g_{I,1}, g_{I,2}, g_{I,3})^T \end{aligned}$$

As the mass matrix (2) can be inverted symbolically, also the matrix $A(\vec{x})$ in (8) can be inverted symbolically, so that the so-called semi-linear implicit state space description (8) can be made explicit:

$$\dot{\vec{x}} = A(\vec{x})^{-1} \cdot \vec{g}(\vec{x}) = \vec{f}(\vec{x}) \quad (9)$$

The explicit state space description (9) can now directly be inserted in the algorithm for the Euler solver (6) or Heun solver (7), if the limitations for the servo motor currents due to (3) are not necessary (otherwise for the limitations a special implementation must be used). The simple Euler solver (6) is of approximation order 1 and has a limited area of stability, so that an appropriate small step size must be used. The Heun solver (7) – which starts with an Euler step and improves the result by the trapezoidal rule – is of approximation order 2 and has also a limited area of stability, but allowing slightly bigger step sizes than Euler solver.

As conclusion, *Task A – Implicit Model Handling* is performed by transformation into an explicit state space description for proper use with explicit ODE solvers.

2 Implementation of Point-to-point Motion

The second task *Task b - Point-to-Point Movement* requires a proper implementation and simulation in the time domain for a point-to-point movement of the robot arm. This benchmark study compares an EXCEL implementation and a MATLAB implementation based on explicit Euler solver and explicit Heun solver using the explicit state space description (9), with modifications for the limitations of the currents. For better comparison, the algorithmic formulations in EXCEL and in MATLAB are as 'near' as possible, and no EXCEL macro features and no MATLAB modules are used.

2.1 Euler implementation – EXCEL

EXCEL is no simulator, it does not provide ODE solvers or other simulation tools. But simple ODE solvers can be easily implemented by means of recursive cell update in columns.

Figure 9 (see last section) shows parts of the spreadsheet implementation. There, the first row denotes time and states `t q1dot q2dot q3dot q1 q2 q3 I1 I2 I3` in columns `P Q R S T U V Z AA AB`, and the second row contains the initial values – all zero.

The following rows are recursive updates for time $t_{i+1} = t_i + h$ in column `P`, and due to (6) Euler integration $x_{i+1} = x_i + h \cdot f(x_i)$ in columns `P Q R S T U V Z AA AB` for the states $\dot{q}_1, \dot{q}_2, \dot{q}_3, q_1, q_2, q_3, I_1, I_2, I_3$, in EXCEL notation for instance given

for time t

```
P3:  = P2 + h
P4:  = P3 + h
P5:  = P4 + h
...  ...
```

for state $\dot{q}_1$

```
Q3:  =Q2+h*f1(Q2 R2 S2 T2 U2...)
Q4:  =Q3+h*f1(Q3 R3 S3 T3 U3...)
Q5:  =Q4+h*f1(Q4 R4 S4 T4 U4...)
...  ...
```

for state q_1

```
T3:  =T2+h*f4(Q2 R2 S2 ...)=T2+h*Q2
T4:  =T3+h*f4(Q3 R3 S3 ...)=T3+h*Q3
T5:  =T4+h*f4(Q4 R4 S4 ...)=T4+h*Q4
...  ...
```

Here the formula functions `f1,f2,...,f9` correspond to the derivative vector $(f_1, f_2, ..., f_9)^T$. `f1,f2,f3` are relatively complicated, as they result from symbolic inversion of the mass matrix (2) – they need auxiliary variables for simplification (see Figure 10, last section); `f4,f5,f6` are trivial, as they are only integrating the velocities (see above); and `f7,f8,f9` are complicated because of the state limitations due to (3) and (4) – the classical problem of space variables which must be limited. The following code snippet shows the EXCEL formula for the Euler update of $x_1 = \dot{q}_1$:

```
q1dot,i = q1dot,i-1+h*
(-ma22/(ma12,i*ma21,i-ma11,i*ma22)*
(u_1*3^0,5/2*kt_1*Ia1,i-1+Th_2*
(2*q1dot,i-1*q2dot,i-1+q2dot,i-1^2)*
SIN(q2,i-1))+ma12,i/
(ma12,i*ma21,i-ma11,i*ma22)
*(u_2*3^0,5/2*kt_2*Ia2,i-1-Th_2*
q1dot,i-1^2*SIN(q2,i-1)))
```

The index `i` hereby implies the time step, the values for these variables have to be calculated for each time step in the EXCEL sheet – in the EXCEL formulas all variables with a time index are replaced with the respective cell name. Variables without an index are constant and are named cells.

In order to implement the case distinction for the current of the motors an interim result for the current is calculated, based on the ODE and the limitations for current and voltage due to (3) and (4). These interim calculations allow to limit the integration on the derivative, instead on the output: Since these interim results could result in values above the given threshold for these quantities an `IF`-statement is used to check if they are above their maximal or below their minimal values. If the interim results are outside the allowed region the given boundary values are used for the calculations, and if they are within the allowed region the interim values are used – see following code snippet:

```
U1,i = P_1*(q1_t-q1,i)-D_1*q1dot,i
U1a,i = IF(ABS(U1,i)>U_1maxreg;
U_1maxreg*SIGN(U1,i);U1,i)
I1,i = I1,i-1+h*((U1a,i-1-kt_1*
u_1*q1dot,i-1-R_a1*Ia1,i-1)/L_a1)
I1a,i = IF(ABS(I1,i)>I_1max;I_1max*
SIGN(I1,i);I1,i)
```

2.2 Euler Implementation – MATLAB

MATLAB is a powerful numerical programming environment for any tasks, also for 'manual' programming of dynamic simulations. SIMULINK is a MATLAB extension for graphical modelling and simulation of dynamic systems based on input/output relations, equipped with a powerful so-called ODE suite with many different ODE solvers – from explicit Euler solver to implicit stiff system solver with state event detection.

In MATLAB only a subset of this ODE suite is available, which does not include Euler solver and Heun solver – but both solvers – named ODE1 and ODE2 – can be downloaded from MathWorks as m-file [2]. As basic ODE1 solver and basic ODE2 solver only terminate on time conditions and do not provide state event termination, for the tasks of this benchmark both solvers had to be modified – for *Task b - Point-to-Point Movement* using a simple `IF`-statement for terminating the integration loop (state update loop) as soon as a certain state is reached. For *Task c - Collision Avoidance* more complex `IF`-statements are necessary to distinguish and terminate the different simulation phases.

Generally, ODE solver libraries require precise formulation of the derivative function $\vec{f}(\vec{x})$ and perform the integration steps (the state updates) unconditionally - so also MATLAB's ODE suite does. That means that derivatives can be limited in the formulation of the function, but states cannot be limited directly.
In case of the SCARA robot, the state variables for the current must be limited in advance, which requires a modification of the integration step. Consequently, also for the MATLAB Euler integration it is needed to make use of `IF`-statements with logic queries to implement the equations of the current and the voltage of the motors in MATLAB. These `IF`-statements are embedded within the ode solver's `FOR`-loop.

First results for the point-to-point motion of the tool tip in 3D space using the Euler solver with a step size of 0.0004 show a reliable behaviour (see Figure 2). The kinematic restrictions result in a bend of the path toward the end and the target position for q_1 and q_2 is reached before that of q_3.

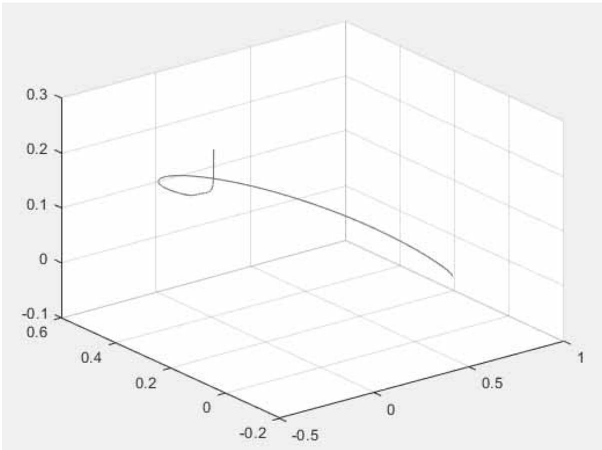


Figure 2: Point-to-point motion of the tool tip in 3D space (Euler solver, MATLAB, step size 0.0004).

2.3 Heun Implementation – EXCEL

The Heun solver requires two evaluations of the derivative function vector $\vec{f}(\vec{x})$, the first $\vec{f}(\vec{x}_i)$ for the Euler approximation (6) by $\vec{x}^E_{i+1} = \vec{x}_i + h \cdot \vec{f}(\vec{x}_i)$, and the second $\vec{f}(\vec{x}^E_{i+1})$ for the Heun correction due to (7).

For both slope vectors, in the EXCEL implementation now auxiliary variables (new columns) are used: `K1_1,...,K1_9` for $\vec{f}(\vec{x}_i)$, and `K2_1,...,K2_9` for $\vec{f}(\vec{x}^E_{i+1})$ (see Figure 10, last section). Using these auxiliary variables, which also double the auxiliary variables for components of the inverted mass matrix, the Heun step for the first state `q1_dot` (being $x_1(t_{i+1})$) becomes:

```
K1,1,i = q1ddot,i-1 = -ma22/
  (ma12,i-1*ma21,i-1-ma11,i-1*ma22)*
  (u_1*3^0,5/2*kt_1*Ia1,i-1+Th_2*
```

```
  (2*q1dot,i-1*q2dot,i-1+q2dot,i-1^2)*
  SIN(q2,i-1))+ma12,i-1/
  (ma12,i-1*ma21,i-1-ma11,i-1*ma22)*
  (u_2*3^0,5/2*kt_2*Ia2,i-1-Th_2*
  q1dot,i-1^2*SIN(q2,i-1))
K2,1,i = q1ddot,pred,i-1 = -ma22/
 (ma12,i-1*ma21,i-1-ma11,i-1*ma22)*
 (u_1*3^0,5/2*kt_1*Ia1,pred+Th_2*
 (2*(q1dot,i-1+h*K1,1,i-1)*
 (q2dot,i-1+h*K1,2,i-1)+
 (q2dot,i-1+h*K1,1,i-1)^2)*
 SIN(q2,i-1+h*q2dot,i-1))+
 ma12,i-1/
 (ma12,i-1*ma21,i-1-ma11,i-1*ma22)*
 (u_2*3^0,5/2*kt_2*Ia2,pred,i-1-Th_2*
 (q1dot,i-1+h*K1,1,i-1)^2*
 SIN(q2,i-1+h*q2dot,i-1))
q1dot,i = q1dot,i-1+(h/2)*
 (K1,1,i-1+K2,1,i-1)
```

In the above EXCEL formula, the last two lines represent the Heun update due to (7).
Predicted values for limited variables as currents and voltages are calculated separately within their boundaries, so in above formula no conditional statements are necessary (but again the number of auxiliary variables represented in new columns increases):

```
U1pred,i = P_1*
 (q1_t-(q1,i+h*K1,1,i))-
 D_1*(q1dot,i+h*K1,1,i)
U1a,pred,i = IF(ABS(U1,pred,i)>
 U_1maxreg;U_1maxreg*
 SIGN(U1,pred,i);U1,pred,i)
I1a,pred,i = IF(ABS(I1,i+h*K1,1,i)>
 I_1max;I_1max*SIGN(I1a,i+h*K1,1,i);
 I1a,i+h*K1,1,i)
```

Despite the mathematical simplicity of the Heun method the necessity of many auxiliary variables the complexity of the calculations in EXCEL increases. A better, but advanced EXCEL technique would be the use of EXCEL macros for the derivative functions.

2.4 Heun Implementation – MATLAB

The implementation of the Heun method in MATLAB directly follows the Euler implementation, but using two evaluations of the derivative function with following use of trapezoidal rule due to (7) in the state update loop. Again the provided Heun solver has to be modified with respect to the state limitations for the currents – with formula very similar to the above sketched EXCEL formula.

3 Obstacle Avoidance

The third task *Task c - Collision Avoidance* requires extension of the model description for handling a collision avoidance manoeuvre. The obstacle, a box, is situated at a certain x-position x_{obs}, and has a certain height h_{obs}. If the tool tip of the robot gets too near to the obstacle in the xy-plane (nearer than a critical distance d_{crit}), motion in xy-plane must stop, and the robot can move only upwards in z-direction (q_3-direction) as fast as possible, until the height of the obstacle is reached. This collision avoidance manoeuvre is given by condition formula

$$(d = x_{tip} - x_{obs}) \leqslant d_{crit} \ \wedge \ q_3 < h_{obs} \qquad (10)$$

3.1 Euler and Heun Implementation – EXCEL

For EXCEL implementation, same the principles as in *Task b - Point-to-Point Movement* are used, but with more complicated control actions depending on conditions. Consequently, several *IF*-statements as well as new auxiliary variables are added to implement the collision avoidance manoeuvre, including Cartesian coordinates for the positions.

During each step the distance d in x-direction between the tool tip and the obstacle due to (10) is calculated. As soon as this distance is smaller than the critical distance and the tool tip is not above the obstacle height target x-position and target y-position are set to the current x-position and current y-position, as well as the boundaries for voltages of the motors are changed to the emergency maximum. This new target position as well as voltage maxima are kept until the tool tip has risen above the obstacle height.

To realize the switch of the target position to the current position an auxiliary variable `d_mod` is calculated which is equal to the actual distance to the obstacle as long as it is bigger than the critical distance, but frozen after the distance falls below the critical value and the tool tip is below the obstacle height. By referencing to this `d_mod` and to the current z-position of the tool tip, the variable boundaries can be changed comfortably to their emergency maximums after the distance falls below the critical distance, and all other calculations are taken from formulas used in the implementation for *Task b - Point-to-Point Movement*. The EXCEL formulas for this controlling d_mod are:

```
dmod,i = IF(AND(dmod,i-1<
 d_crit;q3,i-1-h_obs<0);
 d_mod,i-1;d,i)
q1_t,i = IF(AND(q3,i-h_obs<0;
 d_mod,i<d_crit);q1,i-1;q1_t,i-1)
U1a,i = IF(AND(q3,i-h_obs<0;
d_mod,i<d_crit);IF(ABS(U1,i)>
U_1max;U_1max*SIGN(U1,i);U1,i);
IF(ABS(U1,i)>U_1maxreg;U_1maxreg*
SIGN(U1,i);U1,i))
```

The method of freezing d_mod is chosen to avoid switching back to the original target position before the tool tip has reached a height above the obstacle and thereby oscillating around the critical distance.

The Heun implementation in EXCEL follows the Euler implementation, but using for state update the Heun solver with additional variables as given in *Task b - Point-to-Point Movement*.

3.2 Euler and Heun Implementation – MATLAB

The MATLAB implementation chooses a separation of the dynamics in three phases: PD-controlled movement to target position until obstacle detection, obstacle avoidance movement until non-critical heights, and PD-controlled movement to target position. For the first phase, the ODE1 solver from *Task b - Point-to-Point Movement* is used, extended by movement stop at obstacle detection due to (10); the second phase is governed by an modified ODE1 solver which uses the (simpler) collisions avoidance control, until a non-critical height is reached, and the third phase can make use of the ODE1 solver from *Task b - Point-to-Point Movement*. In MATLAB all results are concatenated:

```
y1 = ode1_1(@(t,y) reach_target(t,y),
     tspan,y0);
y2 = ode1_2(@(t,y) avoid_obs(t,y),
     tspan,y1(length(y1),:));
y3 = ode1_3(@(t,y) reach_target(t,y),
     tspan,y2(length(y2),:));
```

In the implemented ODE solvers IF-statements (given below) stop the integration loop on occurence of a specific event. The first IF-statements stops the integration loop of ODE1_1 if the tool tip drops below the security threshold due to (10). The second IF-statements stops the integration loop of ODE1_2, if the tool tip has exceeded the obstacle heights, and the integration loop of ODE1_3 stops when the tool tip reaches the target position:

```
if  1 - (abs(L1*cos(Y(4,i+1)) +
  L2*cos(Y(4,i+1)+Y(5,i+1)) - xobs)
  <=dcrit && (Y(6,i+1)<hobs)) == 0
if  (1 - ((Y(6,i+1) - hobs)>hsafe)) == 0
if  (abs(Y(4,i+1) - 2) < 0.001
  & abs(Y(5,i+1) - 2) < 0.001
  & abs(Y(6,i+1) - 0.3) < 0.001) - 1 == 0
```

To detect the obstacle, the tool tip position is calculated and an IF-statement checks whether the position exceeds any restriction. Afterwards when the event is detected the position of the tool tip is locked in x-direction and y-direction, and the second ode solver started.

To detect the end of the obstacle, another IF-statement compares the current tool tip height with the obstacle height and stops the solver as soon as the tool tip exceeds the obstacle height. Thereafter the last ODE solver takes over and lets the robot move freely until the tool tip arrives at the designated position.

The Heun implementation simply replaces the modified Euler ODE1 solver by the modified Heun ODE2 solver for construction the three different Heun solvers ODE2_1, ODE2_2, and ODE2_3 for the three phases.

4 Results – Comparison – Discussion

In order to compare the different solutions between MATLAB and EXCEL on the one side, and between Euler solver and Heun solver, all simulations are performed with the same step size h. Results from EXCEL simulations are imported into MATLAB and plotted with MATLAB plot features.

The choice of a proper step size is a critical task. Euler solver and Heun solver are explicit solvers, so they have limited stability regions, and so the step size is also limited – on the other hand side small step sizes result in a very big number of rows in the EXCEL implementations - a minimum of 3000 in the calculated simulations.

4.1 Results point-to-point motion

The reliable results in Figure 2 for the point-to-point motion of the tool tip in 3D are calculated with a step size of $h = 0.0004$. This step size requires about 7000 rows in EXCEL.

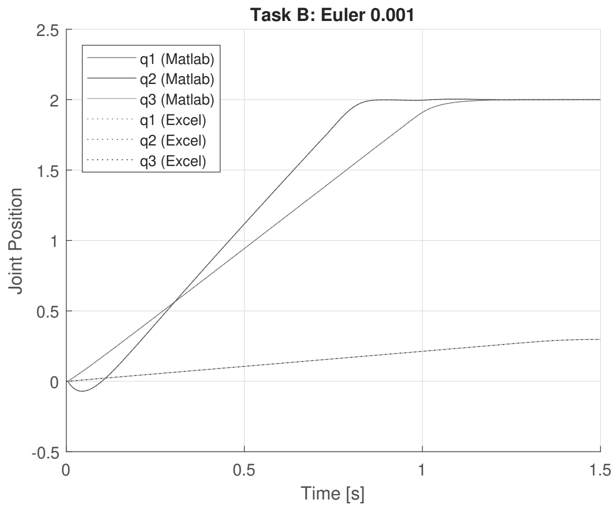


Figure 3: States q_1, q_2 and q_3 over time for *Task b - Point-to-Point Movement*, Euler solver with step size 0.001. EXCEL solutions (dotted lines) and MATLAB solutions (solid lines) show negligible differences.

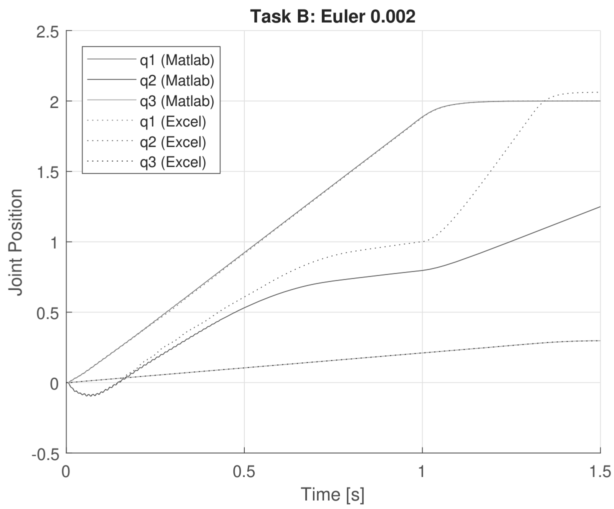


Figure 4: States q_1, q_2 and q_3 over time for *Task b - Point-to-Point Movement*, Euler solver with step size 0.002. EXCEL solutions (dotted lines) and MATLAB solutions (solid lines) differ significantly.

Time domain results for the joint coordinates with same step size $h = 0.0004$ and step size up to to step size of $h = 0.001$ (only 3000 rows necessary) coincide for EXCEL and for MATLAB implementation – 'classical' correct results for this benchmark, as given in Figure 3.

Experiments with the step size in *Task b - Point-to-Point Movement* indicate, that the step size $h = 0.001$ is a 'critical' maximal allowable step size. Figure 3 compares the MATLAB results and the EXCEL results for this step size $h = 0.001$ and shows graphically a good coincidence; also a numerical comparison results in a minor expected deviation.

But for a step size of $h = 0.002$ and bigger the solutions differ more significantly, as documented in Figure 4 graphically, and as checks of the numerical differences proof. Interestingly, the differences of the EXCEL solutions with $h = 0.001$ and $h = 0.002$ are bigger than the differences of the MATLAB solutions with $h = 0.001$ and $h = 0.002$. A possible reason is a more sensitive behaviour of EXCEL due to accumulating round-off errors – a topic for further investigation and better error parameter tuning in EXCEL.

Usually the use of a higher order ODE solver lets expect more accurate results with the same step size, or results with same accuracy using a bigger step size. Unfortunately this expectation does not hold for the Heun solver in case of the investigates model, although he has order 2. The stability region of the Heun solver extends only in the imaginary direction for the eigenvalues. This would allow bigger step sizes for oscillating behaviour, being not the case in the investigated model. Consequently also for the Heun solver the step size $h = 0.001$ is the critical maximal possible step size. A bigger step $h = 0.002$ results in differences similar to that of the Euler solver with step size $h = 0.002$, and additionally the EXCEL solutions are more stronger affected by round-off errors, so that EXCEL results with Heun and step size $h = 0.002$ are worse than EXCEL results with Euler and step size $h = 0.002$, especially EXCEL results for state q_2 seem to be definitely wrong.

4.2 Results collision avoidance

Generally, the results for *Task c - Collision Avoidance* are reliable for the MATLAB implementation and for the EXCEL implementation, if the step size is chosen properly.

Figure 5 displays the results for the tool tip position x_{tip} and for $q_3 - h_{obs}$, the distance to the obstacle in z-direction over time for Euler solver with step size $h = 0.001$: the tool tip is approaching the obstacle, after detection stopping movement in x-direction (also moving 'back' a little), and continuing x-direction movement after reaching the security height. The solid lines for the MATLAB solutions overlap the dotted lines for the EXCEL solutions, as the results are almost the same. These results are similar to other benchmark solutions already published, with slight differences at begin of the collision avoidance manoeuvre because of differences in implementing the manoeuvre.

Again the step size $h = 0.001$ turns out to be the maximal allowable one. Euler solver with step size $h = 0.002$ results in differences between MATLAB implementation and EXCEL implementation for the tool tip position x_{tip} already in the first phase (approaching the obstacle), increasing in the phase of collision avoidance manoeuvre, and couriously overshooting in the third phase (Figure 6). The use of the Heun solver does not improve the accuracy, in contrary: the deviations between MATLAB implementation and EXCEL implementation for step size $h = 0.002$ are worsening.

For completeness, Figure 7 shows the motion of the tool tip in 3D space. There, due to the momentum of the system the tool tip overshoots the critical distance at first and then returns to the newly set target position in the xy-plane resulting in a slight bend of the motion next to the obstacle,

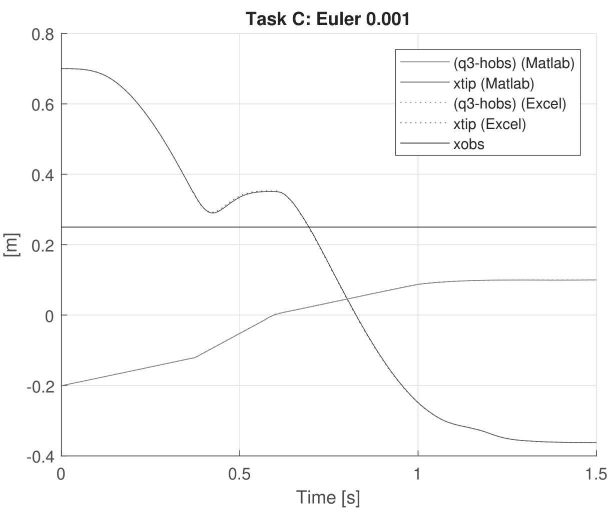


Figure 5: Tool tip position in x-direction x_{tip} and distance to the obstacle in z-direction $q_3 - h_{obs}$ for *Task c - Collision Avoidance*, calculated with Euler solver and step size $h = 0.001$.MATLAB and EXCEL solutions are almost congruent.

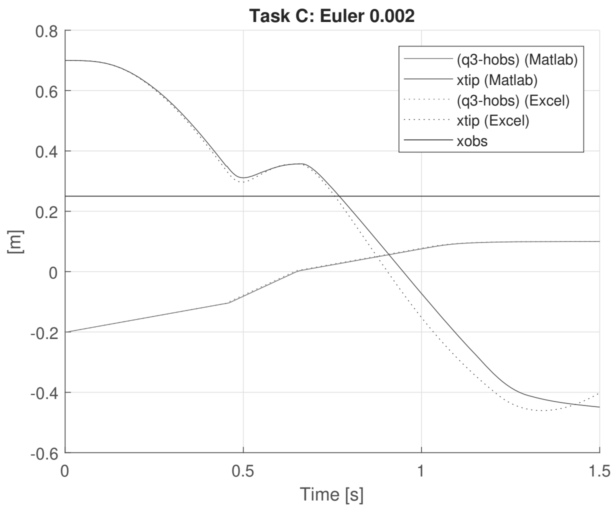


Figure 6: Tool tip position in x-direction x_{tip} and distance to the obstacle in z-direction $q_3 - h_{obs}$ for *Task c - Collision Avoidance*, calculated with Euler solver and step size $h = 0.002$. MATLAB solutions and EXCEL solutions differ, the EXCEL solution shows a slight overshoot.

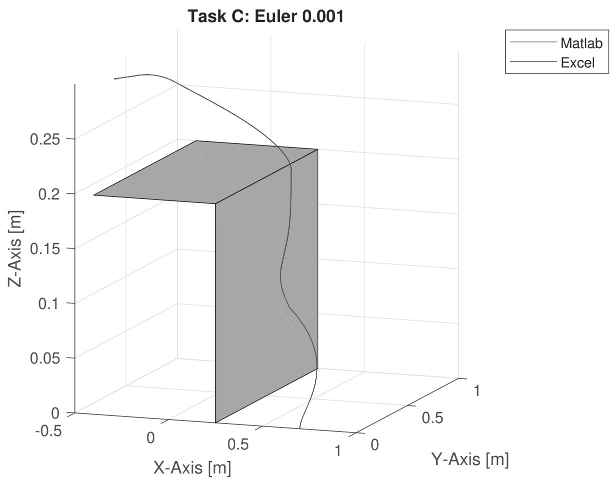


Figure 7: Motion of the tool tip in 3D space for *Task c - Collision Avoidance* using Euler solver with a step size $h = 0.001$. Due to the momentum of the system the tool tip overshoots the critical distance at first and then returns to the newly set target position in the xy-plane resulting in a slight bend of the motion next to the obstacle.

4.3 Comparison MATLAB - EXCEL

MATLAB is a powerful numerical programming environment for any tasks, also for 'manual' programming of dynamic simulations. SIMULINK is a MATLAB extension for graphical modelling and simulation of dynamic systems based on input/output relations, equipped with a powerful so-called ODE suite with many different ODE solvers – from explicit Euler solver to implicit stiff system solver with state event detection.

In MATLAB only a subset of this ODE suite is available, not including Euler solver and Heun solver, and not offering features for event detection and limited integration. So in any case the limitations, event detection, and event actions must be programmed 'manually' - with `IF-THEN-ELSE`-constructs – similar to the implementation in EXCEL. So basic MATLAB is not the best tool for the tasks of *ARGESIM Benchmark C11 'SCARA Robot'*.

A spreadsheet tool as EXCEL is definitely not a simulator – modelling features for ODEs, processes, events, etc. are missing. But spreadsheet programs are an excellent experiment environment with statistical analysis, optimisation, what-if analysis, data handling, etc. Of course, macros and external programming could be used, but to some extent the standard features allow to implement explicit ODE solvers as recursive formulas.

Basic implementations are faced with the problem of equidistant small step sizes, which are necessary in case of technical dynamic systems; here the round-off errors cause problems, and the number of rows in the spreadsheet may increase drastically.

It is to be noted, that also variable step size control could be implemented: in case of using solvers with different order (as here with Euler and Heun) the difference of the solvers in the integration step estimates the error, so that in case of a too big error the step size can be decreased – and increased in case of very small error. Especially the second case - step size increase - could prevent from the EXCEL-genuine round-off error.

But a general disadvantage is the lack of accuracy in the EXCEL standard configuration – possible but laborious to improve. On the other hand, a spreadsheet tool is a very suitable tool for education, so that this C11 benchmark study is mainly intended for educational use. On the other side, the use of advanced EXCEL features as macros, programmed modules, and EXCEL add-ons would allow much more comfortable implementation and also more accuracy.

References

[1] Horst Ecker *Comparison 11: SCARA Robot.* EUROSIM-Simulation News Europe, Number 22; March 1998. 30-32

[2] MathWorks Support Team *Is there a fixed-step Ordinary Differential Equation (ODE) solver in MATLAB 8.0 (R2012b)*; https://de.mathworks.com/matlabcentral/answers/98293; Oct. 2012

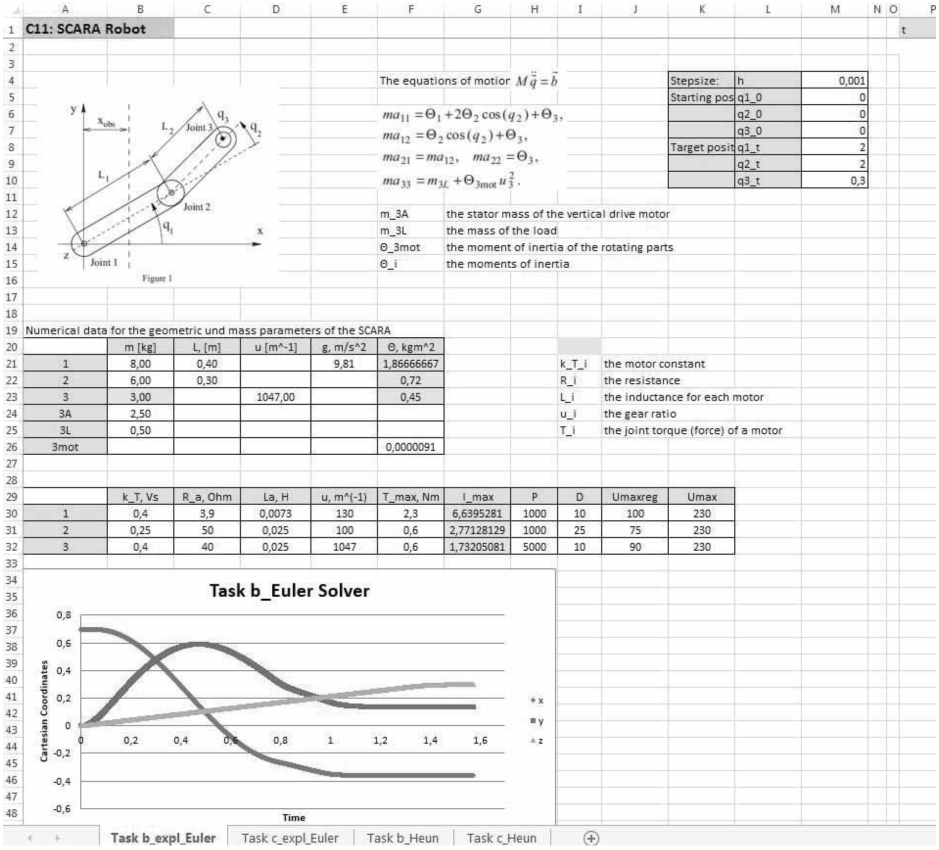
5 Appendix

EXCEL is no simulator, it does not provide ODE solvers or other simulation tools, and it does not have a structure for implementing dynamic models. On the other hand, EXCEL allows calculation and documentation of any kind and in any structure. So also ODE solvers - being state updates - can be implemented by recursive formula in consecutive rows (and cells).

The implementation developed in this benchmark study provides different worksheets for tasks and ODE solvers, but with same structure, see Figure 8. The upper left region of all worksheets (columns `A,B,..M`) is reserved for definition and documentation of the system: model sketch, summary of equations, definition of parameters (named cells), etc. Furthermore, right above the simulation parameters can be put in and changed: initial and target position, and step size for the ODE solver. At bottom, time diagrams are provided.

The calculation area starts with column P. Figure 9 sketches the first five rows of the recursive implementation of the Euler solver in columns `P, Q, ..,AB`. There, the first row denotes time and states, the sec-ond sets the initial values, and the following rows calcu-late recursively updates for time $t_{i+1} = t_i + h$ and states $x_{i+1} = x_i + h \cdot f(x_i)$ due to (6) Euler integration - details see Section 2.1. Depending on step size and on distance to target, a usually big number of rows have to be used for the full time course.

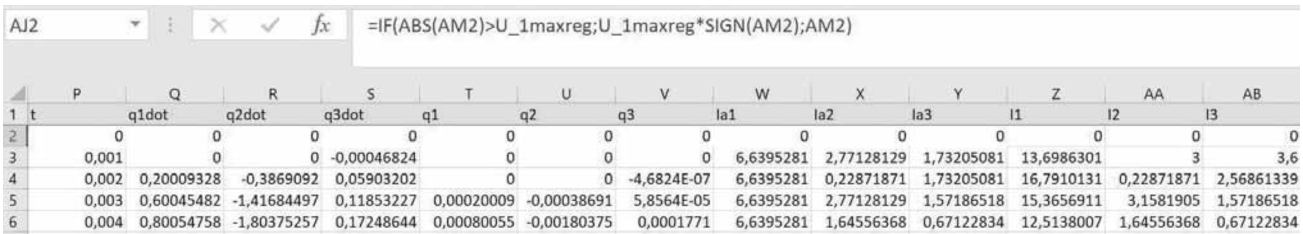
For calculating the derivative functions, the following columns `AC,...,AN` provide auxiliary and control variables. The Heun solver must calculate a second evaluation of the derivative functions, so further columns from column AR on are foreseen (details in Section 2.3, sketch in Figure 10). The cell content window in Figure 9 and Figure 10 show the formula for calculation the control voltage: from a relatively simple formula in Figure 9 for *Task b - Point-to-Point Movement* to a more complex one for *Task c - Collision Avoidance* in Figure 10.



	m [kg]	L [m]	u [m^-1]	g, m/s^2	Θ, kgm^2
1	8,00	0,40		9,81	1,86666667
2	6,00	0,30			0,72
3	3,00		1047,00		0,45
3A	2,50				
3L	0,50				
3mot					0,0000091

	k_T, Vs	R_a, Ohm	La, H	u, m^(-1)	T_max, Nm	I_max	P	D	Umaxreg	Umax
1	0,4	3,9	0,0073	130	2,3	6,6395281	1000	10	100	230
2	0,25	50	0,025	100	0,6	2,77128129	1000	25	75	230
3	0,4	40	0,025	1047	0,6	1,73205081	5000	10	90	230

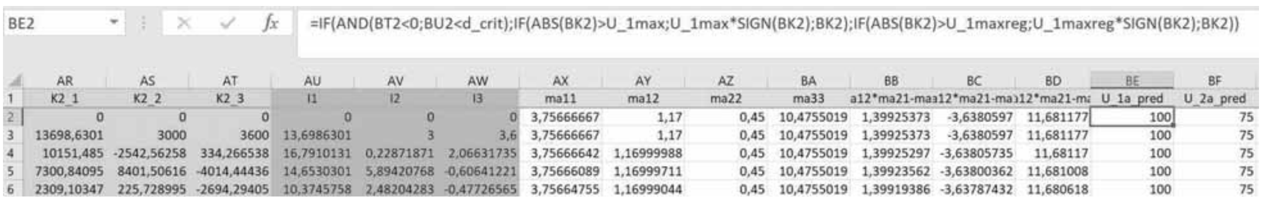
Figure 8: EXCEL implementation – definition and documentation area.



AJ2 =IF(ABS(AM2)>U_1maxreg;U_1maxreg*SIGN(AM2);AM2)

	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
1	t	q1dot	q2dot	q3dot	q1	q2	q3	la1	la2	la3	I1	I2	I3
2	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0,001	0	0	-0,00046824	0	0	0	6,6395281	2,77128129	1,73205081	13,6986301	3	3,6
4	0,002	0,20009328	-0,3869092	0,05903202	0	0	-4,6824E-07	6,6395281	0,22871871	1,73205081	16,7910131	0,22871871	2,56861339
5	0,003	0,60045482	-1,41684497	0,11853227	0,00020009	-0,00038691	5,8564E-05	6,6395281	2,77128129	1,57186518	15,3656911	3,1581905	1,57186518
6	0,004	0,80054758	-1,80375257	0,17248644	0,00080055	-0,00180375	0,0001771	6,6395281	1,64556368	0,67122834	12,5138007	1,64556368	0,67122834

Figure 9: EXCEL implementation – calculations for *Task b - Point-to-Point Movement*.



BE2 =IF(AND(BT2<0;BU2<d_crit);IF(ABS(BK2)>U_1max;U_1max*SIGN(BK2);BK2);IF(ABS(BK2)>U_1maxreg;U_1maxreg*SIGN(BK2);BK2))

	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC	BD	BE	BF
1	K2_1	K2_2	K2_3	I1	I2	I3	ma11	ma12	ma22	ma33	a12*ma21-ma	12*ma21-ma	12*ma21-ma	U_1a_pred	U_2a_pred
2	0	0	0	0	0	0	3,75666667	1,17	0,45	10,4755019	1,39925373	-3,6380597	11,681177	100	75
3	13698,6301	3000	3600	13,6986301	3	3,6	3,75666667	1,17	0,45	10,4755019	1,39925373	-3,6380597	11,681177	100	75
4	10151,485	-2542,56258	334,266538	16,7910131	0,22871871	2,06631735	3,75666642	1,16999988	0,45	10,4755019	1,39925297	-3,63805735	11,68117	100	75
5	7300,84095	8401,50616	-4014,44436	14,6530301	5,89420768	-0,60641221	3,75666089	1,16999711	0,45	10,4755019	1,39923562	-3,63800362	11,681008	100	75
6	2309,10347	225,728995	-2694,29405	10,3745758	2,48204283	-0,47726565	3,75664755	1,16999044	0,45	10,4755019	1,39919386	-3,63787432	11,680618	100	75

Figure 10: EXCEL implementation – calculations for *Task c - Collision Avoidance*.

SNE Simulation News

EUROSIM Data and Quick Info

Contents

Simulation Notes Europe SNE is the official membership journal of **EUROSIM** and distributed / available to members of the **EUROSIM Societies** as part of the membership benefits.

If you have any information, announcement, etc. you want to see published, please contact a member of the editorial board in your country or the editorial office. For scientific publications, please contact the EiC.

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

SNE Reports Editorial Board

SNE Editorial Office /ARGESIM

→ *www.sne-journal.org*, *www.eurosim.info*
office@sne-journal.org, *eic@sne-journal.org*
SNE Editorial Office
Johannes Tanzler (Layout, Organisation)
Irmgard Husinsky (Web, Electronic Publishing)
Felix Breitenecker EiC (Organisation, Authors)
ARGESIM/Math. Modelling & Simulation Group,
Inst. of Analysis and Scientific Computing, TU Wien
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 *Full Members* and *Observer Members*, and member candidates.

ASIM	Arbeitsgemeinschaft Simulation *Austria, Germany, Switzerland*
CEA-SMSG	Spanish Modelling and Simulation Group *Spain*
CSSS	Czech and Slovak Simulation Society *Czech Republic, Slovak Republic*
DBSS	Dutch Benelux Simulation Society *Belgium, Netherlands*
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*
NSSM	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*

Societies in Re-Organisation:

CROSSIM	*Croatian Society for Simulation Modeling* Croatia
FRANCOSIM	*Société Francophone de Simulation* *Belgium, France*
HSS	*Hungarian Simulation Society;* Hungary
ISCS	*Italian Society for Computer Simulation* *Italy*

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, and Treasurer are elected out of members of the board.

President	M. Mujica Mota (DBSS), *m.mujica.mota@hva.nl*
Past President	Emilio Jiménez (CAE-SMSG), *emilio.jimenez@unirioja.es*
Secretary	Nikolas Popper, *niki.popper@dwh.at*
Treasurer	Felix Breitenecker (ASIM) *felix.breitenecker@tuwien.ac.at*
Webmaster	I. Husinsky, *irmgard.husinsky*@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**, was organised by **CEA-SMSG**, the Spanish Simulation Society, in La Rioja, Logroño, Spain, July 1-5, 2019. → *www.eurosim2019.com*

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 400 individual members (including associated), and 90 institutional or industrial members.

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

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

✉ ASIM – Inst. of 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*
	A. Körner, *andreas.koerner@tuwien.ac.at*
Secretary	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
	I. Husinsky, *Irmgard.husinsky@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*
Internat. Affairs – GI Contact	O. Rose, *Oliver.Rose@tu-dresden.de*
	N. Popper, *niki.popper@dwh.at*
Editorial Board SNE	T. Pawletta, *thorsten.pawletta@hs-wismar.de*
	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
Web EUROSIM	I. Husinsky, *Irmgard.husinsky@tuwien.ac.at*

Last data update September 2018

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

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

Furthermore, **ASIM** is co-sponsor of *WSC* - Winter Simulation Conference, of SCS conferences *SpringSim* and *SummerSim*, and of *I3M* and *Simutech* conference series.

ASIM Working Committees	
GMMS	Methods in Modelling and Simulation Th. Pawletta, *thorsten.pawletta@hs-wismar.de*
SUG	Simulation in Environmental Systems Jochen Wittmann, *wittmann@informatik.uni-hamburg.de*
STS	Simulation of Technical Systems Walter Commerell, *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	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

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*). **CSSS** main objectives 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 June 2016

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.

→ *www.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*
Vice President	Egils Ginters, *egils.ginters@rtu.lv*
Secretary	Artis Teilans, *artis.teilans@rta.lv*
Repr. EUROSIM	Egils Ginters, egils.ginters@rtu.lv
Deputy	Artis Teilans, *artis.teilans@rta.lv*
Edit. Board SNE	Juri Tolujew, *Juri.Tolujew@iff.fraunhofer.de*
Web EUROSIM	Vitaly Bolshakov, *vitalijs.bolsakovs@rtu.lv*

Last data update June 2019

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

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 **EUROSIM** Full Member in September 2018.

→ *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

NSSM – National Society for Simulation Modelling (Russia)

NSSM - 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 NSSM has become full member.

→ *www.simulation.su*

yusupov@iias.spb.su

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

NSSM 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@mail.ru*, *senyb@dcn.icc.spbstu.ru*,

Last data update February 2018

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	Simon Tomažič, *simon.tomazic@fe.uni-lj.si*
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 December 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

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/societies/romsim/*

florin_h2004@yahoo.com

ROMSIM / Florin Hartescu,
National Institute for Research in Informatics, Averescu
Av. 8 – 10, 011455 Bucharest, Romania

ROMSIM Officers	
President	N. N.
Vice president	Florin Hartescu, *florin_h2004@yahoo.com* Marius Radulescu, *mradulescu.csmro@yahoo.com*
Repr. EUROSIM	Marius Radulescu, *mradulescu.csmro@yahoo.com*
Deputy	Florin Hartescu, *florin_h2004@yahoo.com*
Edit. Board SNE	Constanta Zoe Radulescu, *zoe@ici.ro*
Web EUROSIM	Florin Hartescu, *florin_h2004@yahoo.com*

Last data update June 2019

ALBSIM – Albanian Simulation Society

The Albanian Simulation Society has been initiated at the Department of Statistics and Applied Informatics, Faculty of Economy at the University of Tirana, by Prof. Dr. Kozeta Sevrani.

The society is involved in different international and local simulation projects, and is engaged in the organisation of the conference series ISTI - Information Systems and Technology. In July 2019 the society was accepted as EUROSIM Observer Member.

→ *www.eurosim.info/societies/albsim/*

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	
Chairt	Kozeta Sevrani, *kozeta.sevrani@unitir.edu.al*
Repr. EUROSIM	Kozeta Sevrani, *kozeta.sevrani@unitir.edu.al*
Edit. Board SNE	Albana Gorishti, *albana.gorishti@unitir.edu.al* Majlinda Godolja, *majlinda.godolja@feut.edu.al*

Last data update July 2019

Societies in Re-Organisation

The following societies are at present inactive or under re-organisation:

- CROSSIM - *Croatian Society for Simulation Modelling*
- FRANCOSIM – Société Francophone de Simulation
- HSS – Hungarian Simulation Society
- ISCS – Italian Society for Computer Simulation

Association Simulation News

ARGESIM is a non-profit association generally aiming for dissemination of information on system simulation – from research via development to applications of system simulation. **ARGESIM** is closely co-operating with **EUROSIM**, the Federation of European Simulation Societies, and with **ASIM**, the German Simulation Society. **ARGESIM** is an 'outsourced' activity from the *Mathematical Modelling and Simulation Group* of TU Wien, there is also close co-operation with TU Wien (organisationally and personally).

→ *www.argesim.org*

→ *office@argesim.org*

→ ARGESIM/Math. Modelling & Simulation Group,
Inst. of Analysis and Scientific Computing, TU Wien
Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria
Attn. Prof. Dr. Felix Breitenecker

ARGESIM is following its aims and scope by the following activities and projects:

- Publication of the scientific journal **SNE** – *Simulation Notes Europe* (membership journal of **EUROSIM**, the *Federation of European Simulation Societies*) – *www.sne-journal.org*
- Organisation and Publication of the A**RGESIM Benchmarks** for *Modelling Approaches and Simulation Implementations*
- Publication of the series **ARGESIM Reports** for monographs in system simulation, and proceedings of simulation conferences and workshops
- Publication of the special series **FBS Simulation** – Advances in Simulation / Fortschrittsberichte Simulation - monographs in co-operation with ASIM, the German Simulation Society
- Organisation of the Conference Series **MATHMOD Vienna** (triennial, in co-operation with **EUROSIM**, **ASIM**, and **TU Wien**) – www.mathmod.at
- Organisation of Seminars and Summerschools on Simulation
- Administration of **ASIM** (German Simulation Society) and administrative support for **EUROSIM** *www.eurosim.info*
- Support of ERASMUS and CEEPUS activities in system simulation for TU Wien

ARGESIM is a registered non-profit association and a registered publisher: **ARGESIM Publisher Vienna**, root ISBN 978-3-901608-xx-y, root DOI 10.11128/z…zz.zz. Publication is open for **ASIM** and for **EUROSIM Member Societies**.

SNE – Simulation Notes Europe

The scientific journal **SNE** – *Simulation Notes Europe* 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** puts special emphasis on the overall view in simulation, and on comparative investigations.
Furthermore, **SNE** welcomes contributions on education in/for/with simulation.

SNE is also the forum for the **ARGESIM Benchmarks** on *Modelling Approaches and Simulation Implementations* publishing benchmarks definitions, solutions, reports and studies – including model sources via web.

→ *www.sne-journal.org*,

→ *office@sne-journal.org*, *eic@sne-journal.org*

→ SNE Editorial Office
ARGESIM/Math. Modelling & Simulation Group,
Inst. of Analysis and Scientific Computing, TU Wien
Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria
EiC Prof. Dr. Felix Breitenecker

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*. Members of **EUROSIM Societies** are entitled to download **SNE** in high-quality, and to access additional sources of benchmark publications, model sources, etc. On the other hand, **SNE** offers **EUROSIM Societies** a publication forum for post-conference publication of the society's international conferences, and the possibility to compile thematic or event-based **SNE Special Issues**.

Simulationists are invited to submit contributions of any type – *Technical Note*, *Short Note*, *Project Note*, *Educational Note*, *Benchmark Note*, etc. via **SNE**'s website:

ASIM ASIM ASIM

ASIM Books – ASIM Book Series – ASIM Buchreihen

Proceedings

Simulation in Production and Logistics 2019 – 18. ASIM Fachtagung Simulation in Produktion und Logistik
M. Putz, A. Schlegel (Hrsg.); 18.-20. 9. 2019, Fraunhofer IWU, Chemnitz); to appear; ASIM AM172.

Tagungsband ASIM SST 2018 - 24. ASIM Symposium Simulationstechnik, HCU Hamburg, Oktober 2018
C. Deatcu, T. Schramm, K. Zobel (Hrsg.), ARGESIM Verlag Wien, 2018; ISBN print: 978-3-901608-12-4; ISBN ebook: 978-3-901608-17-9; 10.11128/arep.56; ARGESIM Report 56; ASIM Mitteilung AM 168

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

Tagungsband ASIM SST 2016 - 23. Symposium Simulationstechnik, HTW Dresden, September 2016
T. Wiedemann (Hrsg.); ARGESIM Verlag Wien, 2016; ISBN Ebook 978-3-901608-49-0; ARGESIM Report 52; ASIM Mitteilung AM 160

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.

Books

Kostensimulation - Grundlagen, Forschungsansätze, Anwendungsbeispiele
T. Claus, F. Herrmann, E. Teich; Springer Gabler, Wiesbaden, 2019; Print ISBN 978-3-658-25167-3; Online ISBN 978-3-658-25168-0; DOI 10.1007/978-3-658-25168-0; ASIM Mitteilung AM 169.

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.

Book Series FBS Simulation – Advances Simulation

Artur Schmidt: **Variantenmanagement in der Modellbildung und Simulation unter Verwendung des SES/MB Frameworks**. FBS 30; ISBN Online 978-3-901608-80-3, ARGESIM Wien, 2019; ISBN print 978-3-903311-03-9, TUVerlag Wien (print on demand), 2019

Gunnar Maletzki: **Rapid Control Prototyping komplexer und flexibler Robotersteuerungen auf Basis des SBE-Ansatzes**. FBS 25; ISBN Online 978-3-901608-75-9, ARGESIM Verlag Wien, 2018; ISBN Print 978-3-903311-02-2, TUVerlag, Wien (print on demand); 2019

Patrick Einzinger: **A Comparative Analysis of System Dynamics and Agent-Based Modelling for Health Care Reimbursement Systems.** FBS 24; ISBN Online 978-3-901608-74-2, ARGESIM Vienna, 2015; ISBN Print 978-3-903311-01-5, TUVerlag Wien (print on demand), 2018

Martin Bruckner: **Agentenbasierte Simulation von Personenströmen mit unterschiedlichen Charakteristiken.** FBS 23; ISBN Online 978-3-901608-73-5, ARGESIM Verlag Wien, 2014; ISBN Print 978-3-903311-00-8, TUVerlag Wien (print on demand), 2018

Stefan Emrich: **Deployment of Mathematical Simulation Models for Space Management.** FBS 22;ISBN Online 978-3-901608-72-8, ARGESIM Publisher Venna, 2013; ISBN Print 978-3-903024-99-1, TUVerlag Wien (print on demand), 2018

Xenia Descovich: **Lattice Boltzmann Modeling and Simulation of Incompressible Flows in Distensible Tubes for Applications in Hemodynamics.** FBS 21; ISBN Online 978-3-901608-71-1, ARGESIM Publisher Vienna, 2012; ISBN Print 978-3-903024-98-4, TUVerlag Wien (print on demand), 2018

Florian Miksch: **Mathematical Modeling for New Insights into Epidemics by Herd Immunity and Serotype Shift.** FBS 20; ISBN Online 978-3-901608-70-4, ARGESIM Publisher Vienna, 2012; ISBN Print 978-3-903024-21-2, TUVerlag Wien (print on demand), 2015

Shabnam Tauböck: **Integration of Agent Based Modelling in DEVS for Utilisation Analysis: The MoreSpace Project at TU Vienna.** FBS 19; ISBN Online 978-3-901608-69-8, ARGESIM Publisher Vienna, 2012; ISBN Print 978-3-903024-85-4, TUVerlag Wien (print on demand), 2015

Christian Steinbrecher: **Ein Beitrag zur prädiktiven Regelung verbrennungsmotorischer Prozesse.** FBS 18; ISBN Print 978-3-901608-68-1, ISBN Online 978-3-901608-99-5, ARGESIM Verlag Wien, 2010

Olaf Hagendorf: **Simulation-based Parameter and Structure Optimisation of Discrete Event Systems** FBS 17; ISBN Print 978 3-901608-67-4, 978-3-901608-98-8 (online), ASIM/ARGESIM Vienna, 2010.

Download via ASIM webpage - **www.asim-gi.org**

www.tuverlag.at

ISBN 978-3-903024-87-8