SNE SIMULATION NOTES EUROPE

Journal on Developments and Trends in Modelling and Simulation

EUROSIM Scientific Membership Journal

Vol. 29 No.4, Dec. 2019

ISSN Online 2306-0271
ISSN Print 2305-9974

DOI 10.11128/sne.29.4.1049
ISBN Print 978-3-903024-97-7

TU Verlag an der Technischen Universität Wien

ARGESIM

EUROSIM Congress and EUROSIM Conferences

EUROSIM CONGRESS 2019

CEA SMSG

July 1-5, 2019, Logroño, La Rioja, Spain → www.eurosim2019.com

Snapshots from EUROSIM Congress 2019 - Report EUROSIM Congress 2019 in SNE 30(1)

Logroño Pilgrim Bridge – the Town's Landmark

Logroño University Campus – Congress Site

Congress Opening – Youngest Participant's Address

Congress Opening

Congress Lunch

EUROSIM Board

SIMS EUROSIM Conference 2020

SIMS Scandinavian Simulation Society

Sept. 22-25, 2020, Oulu, Finland → www.scansims.org

MATHMOD Vienna 2021

asim

Feb. 24-26, 2021, Vienna, Austria → www.mathmod.at

EUROSIM CONGRESS 2022

DBSS Dutch Benelux Simulation Society

July 2022, Amsterdam, The Netherlands → www.dutchbss.org

Editorial

Dear Readers,

SNE's main publication aims are (i) quick publication of submitted papers on new trends and developments in simulation, (ii) post-conference publication of contributions to conferences of EUROSIM societies, and (iii) publication of contributions regarding the ARGESIM Benchmarks for Modelling Approaches and Simulation Implementation. Readers may find in this issue SNE 29(4) all of these publication types.

The issue starts with a very interesting overview note by O. Ullrich and co-authors, who investigate indicator identification and data acquisition in co-operating with municipal partners. A technical note by M. Bachler and J. Straub reports on models for heart rate variability beyond the autonomic nervous system. Three post-conference publications complete the technical notes, the first from the EUROSIM Congress 2019: H. Folkerts and co-authors discuss model specification and automatic model generation for multiple simulators– further EUROSIM 2019 post-conference publications are planned for SNE 30(1) and SNE 30(2).

The contributions from the ASIM Symposium Simulation Technique Hamburg 2018 deal with agent-based simulation and process-interaction (S. Xie et al.), and with dynamic vehicle routing problems (T. Mayer et al.), resp. And last but not least, we are glad to publish the first Benchmark Study for benchmark C22 'Non-standard Queuing Policies' by P. Junglas and T. Pawletta, based on a GPSS-MATLAB implementation – a valuable template for further solutions, reports, and studies for this benchmark.

I would like to thank all authors for their contributions to SNE 29(4) - 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(4)

Online SNE 29(4), DOI 10.11128/sne.29.4.1049
ARGESIM Publisher, Vienna, www.argesim.org
Print SNE 29(4) ISBN 978-3-903024-97-7,
TU Verlag Vienna, Print-on-Demand, www.tuverlag.at

Co-operating with Municipal Partners on Indicator Identification and Data Acquisition.
O. Ullrich, M. Bogen, D. Lückerath, E. Rome 159

Agent-based Simulation with Process-interaction Worldview. *S. Xie, T. Zhang, O. Rose* 169

Simulation-based Evaluation of Dynamic Vehicle Routing Problem Features for Algorithm Selection.
T. Mayer, T. Uhlig, O. Rose 179

Expansion of Models for Heart Rate Variability beyond the Autonomic Nervous System.
J. Straub, M. Bachler 189

Solving ARGESIM Benchmark C22 'Non-standard Queuing Policies' with MatlabGPSS.
P. Junglas, T. Pawletta 199

Python-Based eSES/MB Framework: Model Specification and Automatic Model Generation for Multiple Simulators.
H. Folkerts, T. Pawletta, C. Deatcu, S. Hartmann 207

EUROSIM Societies Short Info .... N1 – N8
EUROSIM 2019 Congress Snapshots .... Cover inside

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.
- **PN** Project Note 6 – 8 p.
- **SW** Software Note , 4 – 6 p.
- **ON** Overview Note – only upon invitation, up to 14 p.
- **EN** Education Note –6 – 8 p.
- **SN** Short Note, max. 6 p.
- **BN** Benchmark Note, 2 – 8 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:

Co-operating with Municipal Partners on Indicator Identification and Data Acquisition

Oliver Ullrich, Manfred Bogen, Daniel Lückerath, Erich Rome

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

SNE 29(4), 2019, 159-168, DOI: 10.11128/sne.29.on.10491
Received: October 31, 2019;
Accepted: November 20, 2019
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna,
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Data acquisition is a significant part of the simulation modeling process. It can be a challenge, especially in larger projects with many participants and stakeholders from different professions: They all bring their individual understanding of data, its availability, formats, and quality, with their respective points of view strongly dependent on their own professional backgrounds. This paper presents a modular process for indicator identification and data acquisition in the context of modeling projects, specifically aimed at researchers or practitioners working with municipal partners, i.e. representatives from city councils, urban planners and administrations, or infrastructure providers. The process consists of five steps: indicator selection, suitability check, data gathering, data quality check, and data management. To further facilitate a clearer understanding of the endeavor, the process defines necessary input and to be expected output artifacts. In addition to presenting the process, the paper describes its examplary execution with partners from the City of Bilbao, Spain, and offers a selection of discovered challenges and lessons learned.

Introduction

In complex multi-disciplinary projects with modeling and simulation components, the acquisition of high-quality data can be a significant challenge. Often, data owners are not modeling experts and without further guidance lack the experience to understand what data is necessary in what form.

Especially in projects where data owners are representatives from city administrations or municipal infrastructure providers, these public servants often have the detailed local knowledge that is needed to develop an effective and efficient model. However, their typically relative low level of experience with the modeling and simulation process means that they can benefit from some extra guidance on the more complex steps of the identification of feasible and valid indicators that quantitatively describe the desired characteristics, and the corresponding acquisition and preprocessing of the necessary data.

This paper describes a process designed to assist modelers in projects with municipal stakeholders with the identification of feasible and expressive indicators and the corresponding acquisition and preparation of the necessary data. The process addresses typical stumbling stones and road blocks, and has been validated in a number of case studies in co-operation with the municipalities of Bilbao, Spain, Greater Manchester, UK, Paris, France, and Bratislava, Slovakia.

The described process has been first developed as part of “Impact and Vulnerability Analysis of Vital Infrastructures and Built-up Areas – IVAVIA” (see [24]), a standardized method for the assessment of climate change-related risks and vulnerabilities in cities and urban environments, based on the well-established approach by the German Federal Ministry for Economic Cooperation and Development (see [12]). IVAVIA was conceived as part of the EU-H2020 project “RESIN – Climate Resilient Cities and Infrastructures” (see [23]) and is further developed in the EU-H2020 project “ARCH – Advancing Resilience of Historic Areas Against Climate-related and Other Hazards” (see [2]).

The paper continues with a short introduction on the background of indicator identification and data acquisition in co-operation with municipal partners (see Section 1) and then presents a systematic process designed to help municipalities with these steps (see Section 2).

The paper then describes the application of the process in the context of modeling climate change-related risks and vulnerabilities for the city of Bilbao, Spain, including the encountered challenges (see Section 3). It then elaborates on some lessons learned in the exemplary execution of the process (see Section 4), and concludes with a short summary and an outlook on the next research steps (see Section 5).

1 Background

An “indicator” is a general concept in statistics, with specialisations in other disciplines (see [25]). In simulation (input) modeling, indicators provide information about states or conditions that are not directly measurable as part of the examined system; they may relate directly or indirectly to the element they are intended to measure. When used in general modeling, the indicator is a variable that contributes to describing the properties of a system. In this context “data” is seen as the specific values of indicators, changing over time or space.

Literature reviews show that data gathering and input modeling is an active field in simulation modeling (see [8]). While more technical introductionary texts on simulation mention data gathering only in passing (see [3] and [30]), Carson (see [5]) describes in his overview on the modeling and simulation process often encountered challenges with data collection, cleansing and analysis: data might not be available or not of the desired quality, unforeseen effort is necessary to clean up databases and files, with clients often not knowing what data they own and whether it is acurate.

A number of authors reported progress at solving the identified issues: Bengtsson et al. (see [4]) introduce processes and tools for examining well-structured systems in the aerospace industry, Perara and Liyanage (see [20]) apply a structured process for batch manufacturing environments, while Skoogh et al. (see [28]) aim to support data management in comparable situations. All these proposals assume well-structured environments with stakeholders from manufacturing or industry that share a certain degree of understanding of computer-based modeling.

Especially in large co-creation projects with many stakeholders who do not share that background, indicator identification and data collection and preparation can necessitate even more significant efforts, with literature suggesting that approximately 30% (see [27]) or more (see [20]) of effort in simulation projects are caused by it. In case local or municipal systems or infrastructure components are subject of the examination, usually at least some of these partners are not modeling experts but representatives of the city administration or local infrastructure providers who lack experience with the modeling and simulation process.

Here, co-creation methods might be useful tools. In co-creation or co-management driven projects methods and models are designed, developed, and tested in close feedback loops with end-users instead of applying a traditional waterfall approach that involves non-experts only at the beginning of the process, in the requirements engineering phase, and at the very end, in the application phase. In large modeling projects with municipal stakeholders applying such techniques helps to avoid the trap of being perceived as being purely science-driven (see [7]) and ensures that project results have high practical applicability and usability values (see e.g. [6], [13], and [22]). Municipal stakeholders are reported to perceive co-creation projects as more perceptive to local issues, with an improved connection to their existing agendas (see [18] and [29]). Especially in the context of the co-management of environmental change and the adaptation to climate change in projects involving municipal partners, co-creation processes have been observed to facilitate the implementation of effective multi-level measures (see [1], [21], and [31]).

2 A Process for Indicator Identification and Data Acquisition

2.1 Input

A planned and structured approach to identify indicators and acquire data in co-operation with municipal partners requires some necessary input. That includes documented results of earlier modeling stages and informal local knowledge, but not large amounts of formalized data.

Main input documents are conceptual models, yet without qualitative attributes, that the modeler will have prepared during earlier stages. These conceptual models (see [5]) might have been prepared using one or more of the usual modeling techniques, resulting in a number of event-driven process chains (see [26]), UML activity diagrams (see [9]), impact chain diagrams (see [14]), or other artifacts.

The need for informal local knowledge is a bit harder to grasp. Here, that knowledge would not only include a deeper understanding of the municipal systems or infrastructure components to be modeled, but also the knowledge of what stakeholders have access to the required information, and whether it is available at the necessary levels of functional, spatial, and temporal resolution. The authors found it helpful to identify at an early stage a project data champion inside the local administration who has informal access to colleagues and representatives from local administration departments and other stakeholders. At this stage, it is important to include all relevant stakeholder groups with the project, and to facilitate a clear understanding of the aims and scope, and the knowledge and data necessary for a successful conclusion. Again, an insider can help to identify all relevant stakeholders.

2.2 Steps

The process consists of five steps to be executed by the modeler in co-creation with municipal stakeholders (see Figure 1).

Select Indicators. In modeling, indicators provide information about states or conditions that are not directly measurable; they may relate directly or indirectly to the element they are intended to represent. Examples of indicators in municipal contexts include:

- the geo-referenced distribution of education levels as an indicator for the degree of receptiveness for complex information on climate change-related issues,
- the geo-referenced distribution of age as an indicator for the vulnerability to heat waves, or
- the distribution of public service interruptions as an indicator for the resilience of infrastructure components.

The values of the selected indicators might later be aggregated with additional components to form a composite indicator. Therefore, the stakeholders need to select at least one indicator for each system state or condition to be represented in the model.

How to start identifying indicators? Established indicator directories can be a great help with identifying suitable indicators. For modeling climate change-related issues in municipal contexts, such directories can be found e.g. in the annex of the Vulnerability Sourcebook (see [12]), the Covenant of Mayors for Climate and Energy Reporting Guidelines (see [17]) or the indicator database of the EU FP7 project “MOVE – Methods for the Improvement of Vulnerability Assessment in Europe” (see [16]). However, because indicators are only useful if the relevant data is available for a local context, a modeler should already start considering existing and necessary data when compiling a set of potential indicators: The best indicator is inoperable if there is no feasible way of acquiring the necessary data. The whole indicator selection and data acquisition activity is an iterative and potentially time-consuming process: identifying an indicator, checking its suitability, gathering data, reformulating the indicator if no suitable data can be found to substantiate it, checking data quality and finding alternative sources where necessary. Subsequently, the described steps should not be considered as isolated parts of a sequential process, but rather as different views of an iterative process.

It is important and advisable to include local expertise, e.g. from technicians and other experts working within the urban area or municipality, who know what data is available. In addition to using the already mentioned indicator directories, it might be worth to examine previous studies for the examined region or city in order to reduce the number of potential indicators for which to check data availability. Finally, it is important to find a consensus between the involved experts and stakeholders on what indicators can and should be employed as part of the simulation modeling project.

Developing a preliminary list of potential indicators creates a significant overhead when it is done the first time. However, a thorough documentated selection process will significantly ease the effort of indicator examination in further iterations of the step and in later modeling projects with comprehensive data demands.

The outputs of this step are:

- Preliminary list and documentation of all potential indicators

Check for Suitability. After potential indicators have been identified, they need to be assessed for their suitability towards the purpose of the model:

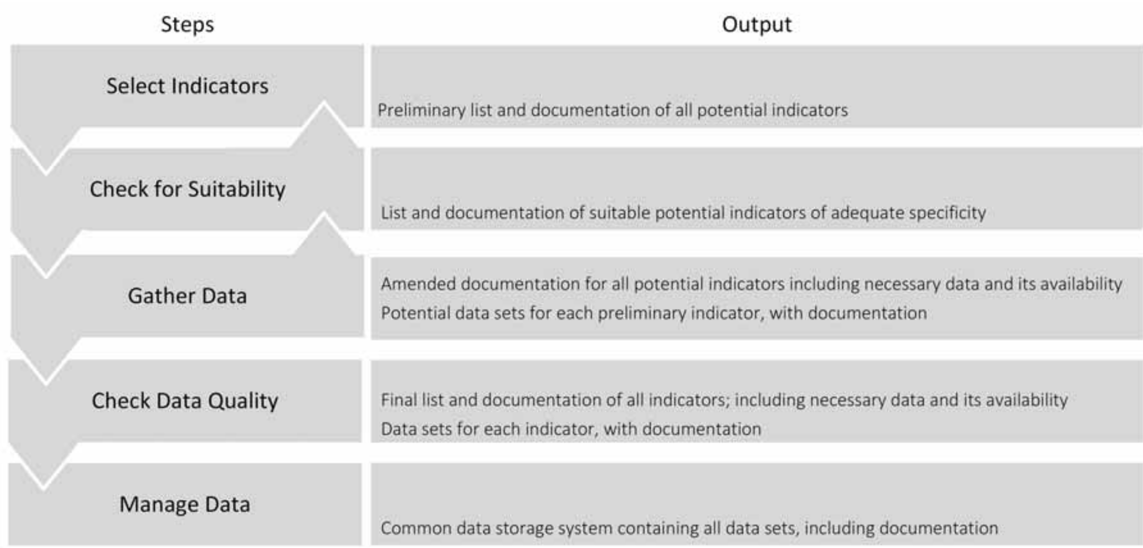

Figure 1: Process steps for indicator identification and data acquisition.

- Are they valid and relevant, i.e. do they represent well the elements to be modelled?
- Are they reliable and credible and allow for data acquisition and measurement in the future?
- Do they have a precise and documented meaning, i.e. do stakeholders agree on what the indicators are measuring in the context of the model?
- Are they clear in their direction, i.e. is an increase in value unambiguously positive or negative?
- Are they practical and affordable, i.e. do they come from accessible data sources?

In an urban or municipal context, two dimensions of appropriateness are often especially relevant:

- Geographical coverage. The identified indicators should cover the full extent of the study area (e.g. the whole urban city or district) and have an appropriate resolution (e.g. population data on a district level).
- Temporal coverage and time frame. Depending on the different indicators identified and whether or not historical periods need to be modeled, archival records and/or future projections need to be available. Additionally, that data should cover the same temporal interval with the same resolution, e.g. daily precipitation and traffic volume for a whole year.

Indicators that are not suitable should be disregarded and their entry in the list of potential indicators amended with a corresponding comment to allow for the reconstruction of the process by non-participating colleagues later on. Should the pruning of the potential indicator list result in elements of the preliminary qualitative model without indicators, the modeler may need to go back to the corresponding step in the process, try to find new indicators and re-iterate the specificity check.

The output of this step is:

- List and documentation of suitable potential indicators of adequate specificity

Gather Data. The process of gathering data for a simulation model of an urban or municipal system can range from extremely simple to highly challenging. It may suffice to download available census data or GIS maps from open access websites. In some cases, time and resources permitting, surveys need to be conducted or large, complex data sets, like satellite images, need to be processed – which might require specialist skills.

What kind of data is needed? There is no standard solution for all possible models. While some indicators will most likely require physically measured or modeled (historical) data, indicators representing social or demographic issues will often require survey and census data or data from expert judgement. However, the most important decision criteria for what data is needed are the study area (e.g. the city as a whole or individual city districts), the desired geographical resolution of the model (e.g. districts, neighbourhoods, smaller areas within neighbourhoods), and the output type (e.g. maps, diagrams, tables) and level of detail (e.g. result validation via visual analysis of the employed data sets).

E.g., if the model represents individual municipal districts, data should at least have a district level resolution (e.g. population data for complete districts or individual households, which then can be assigned to corresponding districts). Data at a lower resolution (e.g. population data for the whole city) usually cannot be disaggregated without loss of validity. As another example, if the simulation results should be represented visually as a map, corresponding geographic data (e.g. a shape file) is needed. Similarly, if detailed (visual) validation of results seems necessary or helpful, the corresponding data needs to be provided in an appropriate format.

Does the data already exist? Based on the information gathered as input for the process (see Section 2.1) a modeler should be able to identify a first set of relevant organizations, facilities, and experts on a local, regional, national, or international level that may be able to provide the necessary data. The huge number of institutions and experts that might need to be contacted can make this one of the most time-consuming steps of the process, especially as follow-up negotiation is often required. Depending on the selected indicators, statistical offices, meteorological authorities, and different national and local government departments might need to be approached. National Spatial Data Infrastructures (NSDI) are another key entry point for data acquisition. NSDIs have been established in many countries and will ideally offer standardised data, even where it is sourced from multiple institutions (see [12]).

When gathering data and contacting institutions and experts, a modeler has to be aware of intellectual property rights levels and sharing policies that may be in place; formal agreements with the respective rights holders could be required.

What resources can be committed to generate data? If data is not available or of insufficient quality, data might need to be collected by the modelers themselves.

In that case, careful assessment of the required costs and expertise needed for the data collection is needed.

Regarding climate-related municipal data, the Vulnerability Sourcebook gives some useful hints (see [12], p. 96-100):

"For meaningful results, observation of biophysical indicators such as precipitation, temperature and run-off must be made over long periods – often over decades. The time and money required for this means it is almost certainly unfeasible for [a modeler to gather this data yourself]. Luckily, however, most countries can provide such data [, e.g. via climate service provides such as hydrometeorological offices]. If you require highly localised data, expert judgement may be a worthwhile alternative."

"Data for socio-economic indicators such as average household income, average size of household and livelihood strategies can be captured in surveys. The time and money required depend largely on the sample size. A representative survey may cover a whole country, or just a few communities. At the sub-national level, surveys can be an effective means of gathering information not captured by national institutions, such as perceptions around climate and environmental change. [In large cities, socio-economic survey data often might have already been collected by a statistics or data department.] Be sure to involve a local expert who can help in drafting the survey, selecting a representative sample and analysing the resulting data."

"Modelled data are both time- and resource-intensive and usually require measured data as input. [...] For meaningful results, you will need to ensure that you can call on the required modelling skills." Often, already existing sub-models for fluvial/pluvial flooding, temperature, and urban heat isles may be available on a regional or national level.

"Where time and financial resources are limited, expert judgement can be a good, fast way of quantifying indicators that cannot otherwise be assessed. This is most often the case at a very local level – [e.g. when gathering information on the municipal sewer system from city technicians] – which is rarely covered by detailed statistical data, and where the climatic and hydrological characteristics are too specific to be captured by modelling. This local knowledge – captured using participative methods as well as scoring and ranking – can be used to either complement or replace surveys."

Given the extent of the data gathering process and its inherent iterative nature, it is perfectly acceptable to conduct the process in an iterative fashion. That is, instead of gathering all potentially necessary data before starting to work on the model itself, it may be more practicable to start with a minimum amount of data necessary. While partial results may lack validity, they may help you when communicating with stakeholders and to secure stronger commitment.

Outputs of this step are:

- Amended documentation for all potential indicators that are specific enough, including what kind of data is needed to quantify the indicator and if it already exists or not
- Potential data sets for each preliminary indicator, with documentation

Check Data Quality. The validity of the output of a simulation model strongly depends on the quality of the input data. Subsequently, a modeler needs to assure that the data gathered is of sufficient quality. Such a check may reveal major issues with the data quality, which may necessitate to repeat at least part of the process. To avoid this, the quality criteria below should already be considered during the data collection. The following paragraphs describe some of the common issues with data quality, both in form and content.

Data format. The collected data can be provided in different formats. To enable easy handling, data should be provided in a well-structured, preferably digital file format that is easy to interpret, does not need manual reformatting, and can potentially be handled by colleagues without the need for specific (software) tools. Literature provides a (non-exhaustive) list of digital file formats that comply with these requirements (see [25]).

Temporal and geographical coverage. The geographical and temporal coverage of different data sources may vary. Thus, the stakeholders need to determine whether different data sources can be combined and compared or not.

In general, data should have the same geographical and temporal coverage. Additionally, data should ideally be as recent as possible and have, or at least be convertible to, the same temporal resolution (e.g. hourly precipitation and traffic volume from June 2017).

However, some differences in data timeliness and temporal coverage may be tolerable. This is especially true if data changes comparatively slowly, e.g. age distribution of citizens or other census data. Additionally, if indicators are independent from one another using data sources with different temporal coverage and resolution may be acceptable. E.g., the amount of green infrastructure in a city and the annual household income will most likely not be related and subsequently a difference in temporal coverage may not be significant.

Another problem may arise if geographical data uses different coordinate systems and projections. This is especially frequent when working on cross-border regions. In order to combine and compare different data sources, a common geographic reference system needs to be employed, such as the Universal Transverse Mercator coordinate system.

In any case, having some data is preferable to having no data at all. If no other alternative to using poor data is available, use it – but be aware that the results of the modeling process may be less reliable or may even be misleading.

Missing values or "outliers". Missing values (e.g. regions omitted from geographical data) can be problematic for simulation modeling. Smaller gaps can be closed with interpolation, i.e. finding existing data nearest to the gaps (in space, time, or function) most likely matching the missing data, or by using the average value as a replacement, if no other data is available. The Handbook on Constructing Composite Indicators by the OECD (see [19]) goes into more detail about missing values and how to deal with them.

"Outliers" on the other hand are values that are so far outside the expected range of the data that they may indicate an error in the capturing or calculation method. While these cases do happen, e.g. from technically unreliable measureing equipment, unexpected data should not be disqualified easily. Are these data points really technical mishaps, or are they a sign that parts of the system to be modeled have not been understood sufficiently?

Outputs are:

- Final list and documentation of all indicators; including what kind of data is needed to quantify the indicator and if it already exists or not
- Data sets for each indicator, with documentation, including physical dimensions

Manage Data. To avoid data loss and lower the risk of data redundancy, all participating stakeholders should store the gathered data sets in a common data storage system. This may range from a simple collection of data files in a set of folders to a more complex database system. Depending on the system employed, they may need to transform the different data sets into a common data format, utilising export and transformation routines from multiple software products. Once such a data storage system is available, its contents can be used for other projects.

In a project with multiple partners and stakeholders it may be necessary to ensure that they can all access the different datasets and work with the same format. Additionally, responsibilities for database management and maintenance might need to be assigned to a "caretaker" or an external service provider. Finally, sensitive data has to be stored in a secure way, only accessible by the appropriate users. What exactly is seen as "sensitive data" is dependent on the country or even region of origin. A good start would be to check the EU General Data Protection Regulation (see [10]) and its national implementation, as well as additional national and regional legislation.

At this point all the gathered data should be documented precisely and comprehensively, allowing all internal and external colleagues, partners, and stakeholders to understand the format and meaning of the data, as well as access rights. Insufficient knowledge about the data can lead to unnecessary duplicate effort from colleagues, data loss, missing results, or a lack in transparency and credibility.

Output of this step is:

- Common data storage system containing all data sets, including documentation, esp. on format and meaning, access rights

2.3 Output

After the successful conclusion of the process steps a modeler has gained a set of documents, including files of actual data, as well as informal knowledge on the to be examined system and its stakeholders. These documents include a final list and documentation of all necessary indicators and the attributes of the system they represent, and a data storage system containing all data sets, including documentation, in the required formats, quality, and granularity.

In addition, a plethora of informal knowledge will have been gained: On one hand, that includes knowledge the modeler gains on the examined system and the municipal stakeholders. The modeler understands which stakeholder has expertise in what specific domain and, should that turn out to be necessary, can provide additional information on a sub-system. On the other hand, the co-operation process will have typically facilitated a better understanding of the modeling and simulation project for the local stakeholders. In the course of the workshops with municipal stakeholders they executed (see Section 3), the authors found that in the best case the project may gain dedicated champions inside the municipal administration.

3 Example

The described process was executed by the authors and representatives from a number of different administrative departments of the City of Bilbao, Spain, as well as other research partners in the context of modeling the impacts of global warming on local residents and infrastructure.

The project started out with an impact chain workshop attended by representatives from different departments where the municipality decided to focus on modeling three cause-impact relationships they thought especially relevant for their city: the impact of extreme precipitation on city traffic infrastructure, the impact of heat waves on public health, and the impact of flooding in built-up areas.

It was clear early in the project that to successfully model these relationships in a meaningful way substantial amounts of data would be necessary that up until then were kept in separate information silos in the different branches of the municipal government. It was thus decided to test the described process in the Bilbao context.

Additional details on the assessment conducted in the Bilbao urban context can be found in [15].

3.1 Identifying Indicators and Acquiring Data for Bilbao

The process of gathering data for the quantitative modeling stages was initiated in parallel to the finalization of the qualitative impact chains. After the immediate start of indicator selection, the project partners began to check whether corresponding sets of data were available and, over the following months, sought to obtain them by establishing contact with different departments within the municipality. As an intermediate result, the partners prepared a first overview of the data found so far for the selected model components and provided the data in differing formats.

The gathered data allowed the authors to start testing and refining the quantitative models and provide intermediate results and visual aids, e.g. maps and charts. Based on discussions of these some of the planned indicators were rejected. That included indicators that were found to be not suitable to measure the desired attribute as initially hoped for by the municipality, such as measureing road accessibility by the number of available CCTV cameras. It was also found that new or additional data had to be provided than initially envisioned by the city administration, e.g. position and specific characteristics of historical public buildings and position of emergency access ways to underground infrastructure components especially prone to flooding.

Data provided in shape file format was processed using a geographic information system software, e.g. to calculate the surface area of green infrastructure or population density per neighbourhood. Data provided in other formats was either converted to shape files (e.g. maps only available as PDF files), used to create new shape files for more complex calculations (e.g. information about coverage area of emergency services was used to create shape files representing the influence radius and subsequently the percentage of population covered by emergency services per neighbourhood), or used directly for indicator calculation (e.g. for capacity of storm tanks).

Approximately nine months into the project the partners collectively decided on the normalization, weighting, and aggregation methods for the selected data at the neighbourhood scale. At the end of that month, an in-person meeting was held in Bilbao between the modelers and the stakeholders to go over the results. The basic model was completed and presented at a workshop organized by the municipality approximately a year after the start of the city case. Then, the municipal stakeholders decided that the spatial resolution was too coarse for the envisioned use of the model. Therefore, the partners tessellated the available data at a finer spatial resolution: a 25 by 25 meter grid, yielding some 66,000 cells for the entire surface of Bilbao. While this resolution allowed for the examination of single city blocks and even individual buildings, the drawback was the labour-intensiveness of preparing the data that way.

3.2 Encountered Challenges

During the course of the described (and other) test cases a number of common challenges and difficulties were encountered, mainly regarding data availability and quality. These include:

- Data sets that employ unwieldy data formats. E.g., a map may only be available as a PDF file without scale, complicating the calculation of surface area and circumference.
- Data sets describing different attributes of the same geographic element may employ different representations of the geographic element, e.g. two shape files of a geographic information system, one describing road elevation and one describing traffic intensities, may show differences in the geographic representation of the underlying road system.
- Data sets with different spatial and/or temporal resolutions. If multiple data sets have to be combined to calculate an indicator, they should have the same spatial and temporal resolution, i.e. they should cover the same area with the same detail over the same time interval. In some cases, it is possible to alleviate differences in resolution, e.g. data from a data set with a very high resolution may be aggregated to fit lower resolution data sets. However, disaggregating data from a low-resolution data set is usually not possible without loss of validity.
- Using the collected data at a later modeling stage may also reveal that a chosen indicator is, against expectations, not suitable to measure the desired phenomenon, e.g. the number of traffic cameras cannot be employed to measure the accessibility of a certain road segment. In this case, it is necessary to replace the indicator with a more suitable one, and to redo the process at least partially.

4 Lessons Learned

The described co-operation process has been tested in the context of four different city cases (see [15]). During that period a number of lessons have been learned both regarding the co-creation process in general and more technical and data quality issues.

Some general lessons from these assessments are: access to local knowledge is necessary, the process cannot be successfully be implemented by external experts only, the process will not be executed successfully without personal contact, face-to-face meetings with local end-users and external experts are necessary, it is important to ensure a common understanding regarding aims, scope, roles, required data, and terminology, it should be made clear from the beginning that the availability of sufficient resources is necessary, offering a reference set of indicators to the local experts can be beneficial and improves the general quality of results, and acquiring necessary data takes much longer than you think – always! A set of requirements concerning data quality and technical issues were identified, all of which have to be fulfilled to ensure the successful execution of the modeling process further down the road: data has to be complete according to specification; data and documentation have to be made available in a language understood both by local domain experts and the modelers; the semantics have to be clear; the spatial and temporal resolution has to be adequate; data formats have to be standardized and well documented; contradictions have to be resolvable or–preferably–not present; and files must not be defective.

5 Conclusion

This paper introduced a process for indicator identification and data acquisition in co-operation with municipal project partners. The process consists of five steps: indicator selection, suitability check, data gathering, data quality check, and data management, and defines the necessary input and the to be expected output documents. As an example, the execution of the process with partners from the City of Bilbao, Spain, was described, and discovered challenges and lessons learned shared.

The described process can help to more systematically approach modeling projects in large, heterogenous teams, and to facilitate a clearer understanding of the steps necessary to acquire data with project partners and stakeholders. It is designed to be executed successfully by domain experts from municipal authorities, who are typically not modeling experts.

In further research steps the process will be utilized, evaluated, and extended in additional projects with municical stakeholders, e.g. the EU-H2020 projects "FORESEE – Future proofing strategies for resilient transport networks against extreme events" (see [11]) and "ARCH – Advancing Resilience of Historic Areas Against Climate-related and Other Hazards" (see [2]).

Acknowledgement

The authors thank their partners in the ARCH and RESIN consortia for their valuable contributions during the development and test process. This paper is based in part upon work in the framework of the European projects "ARCH – Advancing Resilience of Historic Areas Against Climate-related and Other Hazards" and "RESIN – Climate Resilient Cities and Infrastructures".

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements no. 653,522 and 820,999. The sole responsibility for the content of this publication lies with the authors. It does not necessarily represent the opinion of the European Union. Neither the EASME nor the European Commission are responsible for any use that may be made of the information contained therein.

References

[1] Armitage DR, Plummer R, Berkes F, Arthur RI, Charles AT, Davidson-Hunt IJ, Diduck AP, Doubleday NC, Johnson DS, Marschke M, McConney P, Pinkerton EW, Wollenberg EK. Adaptive co-management for social-ecological complexity. In: *Frontiers in Ecology and the Environment*, Vol. 7, 2009, pp. 95-102.

[2] ARCH – Advancing Resilience of Historic Areas Against Climate-related and Other Hazards, http://www.savingculturalheritage.eu/, accessed on Sept 30, 2019.

[3] Banks J, Carson JS, Nelson BL, Nicol DM. *Discrete-Event System Simulation*. Pearson, 2010.

[4] Bengtsson N, Shao G, Johansson B, Leong S, Skoogh A, Mclean C. Input data management methodology for discrete event simulation. In: *Proceedings of the 2009 Winter Simulation Conference*, 2009, pp. 1335-1344.

[5] Carson JS. Introduction to Modeling and Simulation. In:*Proceedings of the 2005 Winter Simulation Conference*, 2005, pp. 16-23.

[6] Cash D, Adger WN, Berkes F, Garden P, Lebel P, Olsson P, Pritchard L, Young OR. Scale and cross-scale dynamics. In: *Ecology and Society*, Vol. 11, 2006.

[7] Chapman E, Hanania S, Connelly A, Carter J. *Deliverable D4.2 Developing the RESIN tools*. EU H2020 Project RESIN, Freiburg, Germany, 2018.

[8] Cheng R. History of Input Modeling. In: *Proceedings of 2017 Winter Simulation Conference*, 2017, pp. 181-201.

[9] Dumas M, ter Hofstede A. UML Activity Diagrams as a Workflow Specification Language. In: *Proceedings of the 4th Conference on the Unified Modeling Language*, 2001, pp. 76-90.

[10] EU General Data Protection Regulation, https://www.eugdpr.org/, accessed Jan 08, 2019.

[11] FORESEE – Future proofing strategies for resilient transport networks against extreme events, https://ec.europa.eu/inea/en/horizon-2020/projects/h2020-transport/infrastructure/foresee, accessed Dec 05, 2018.

[12] German Federal Ministry for Economic Cooperation and Development: *The Vulnerability Sourcebook. Concept and guidelines for standardised vulnerability assessments*. Deutsche Gesellschaft für Internationale Zusammenarbeit, Bonn and Eschborn, Germany, 2014.

[13] Lemos MC, Morehouse BJ. The co-production of science and policy in integrated climate assessments. In: *Global Environmental Change*, Vol 15, 2005, pp. 57-68.

[14] Lückerath D, Bogen M, Rome E, Sojeva B, Ullrich O, Worst R, Xie J. The RESIN climate change adaptation project and its simple modeling approach for risk-oriented vulnerability assessment. In: *Simulation Notes Europe*, Vol. 28, No. 2, 2018, pp. 49-54.

[15] Lückerath D, Dumonteil M. *Deliverable D2.5 Applying IVAVIA, a quantitative risk-based vulnerability assessment methodology*. EU H2020 Project RESIN, Sankt Augustin, Germany, 2017.

[16] MOVE – Methods for the Improvement of Vulnerability Assessment in Europe, Indicator Database Visualiser, http://www.gi4drr.org/move/, accessed on Oct 25, 2017.

[17] Neves A, Blondel L, Brand K, Hendel Blackford S, Rivas Calvete S, Iancu A, Melica G, Koffi Lefeivre B, Zancanella P, Kona A. *The Covenant of Mayors for Climate and Energy Reporting Guidelines*, EUR 28160 EN, 2016.

[18] O'Brien K, Eriksen S, Nygaard LP, Schjolden A. Why different interpretations of vulnerability matter in climate change discourses. In: *Climate Polic*y, Vol. 7, 2007, pp.73-88.

[19] OECD: *Handbook on constructing composite indicators: methodology and user guide* . Paris: OECD Publishing.

[20] Perara T, Liyanage K. Methodology for rapid identification and collection of input data in the simulation of manufacturing systems. In: *Simulation Practice and Theory*, Vol. 7, No. 7, 2000, pp. 645-656.

[21] Peterson G, De Leo G, Hellmann J, Janssen M, Kinzig A, Malcolm J, O'Brien K, Pope S, Rothman D, Shevliakova E, Tinch R. Uncertainty, Climate Change, and Adaptive Management. In: *Ecology and Society*, Vol. 1, No. 2, 1997.

[22] Reed MS, Stringer LC, Fazey I, Evely AC, Kruijsen JHJ. Five principles for the practice of knowledge exchange in environmental management. In: *Journal of Environmental Management*, Vol. 146, 2014, pp. 337-345.

[23] RESIN – Climate Resilient Cities and Infrastructures, http://www.resin-cities.eu/, accessed Nov 11, 2018.

[24] Rome E, Bogen M, Lückerath D, Ullrich O, Worst R, Xie J. IVAVIA: Impact and Vulnerability Analysis of Vital Infrastructures and Built-Up Areas. In: *Critical Information Infrastructures Security - 13th International Conference*, CRITIS 2018, Revised Selected Papers, Springer Security and Cryptology No. 12,260, 2019, pp. 84-97.

[25] Rome E, Bogen M, Worst R, Lückerath D, Sojeva B, Ullrich O, Voss H, Voss N, Xie J. *IVAVIA Guideline, Annex to Deliverable D2.3 Realisation and implementation of IVAVIA*. EU H2020 Project RESIN, Sankt Augustin, Germany, 2017.

[26] Scheer A-W, Thomas O, Adam O. Process Modeling Using Event-Driven Process Chains (chapter). In: Dumas, M., van der Aalst, M., ter Hofstede, A. (ed.): *Process-Aware Information Systems*, John Wiley & Sons, 2005, pp. 119-145.

[27] Skoogh A, Johansson B. Time-consumption analysis of input data activities in discrete event simulation projects. In: *Proceedings of the Swedish Production Symposium*, Vol. 1, 2007.

[28] Skoogh A, Johansson B, Hanson L. Data Requirements and Representation for Simulation of Energy Consumption in Production Systems. In: *Proceedings of CIRP Conference on Manufacturing System*s, Vol. 44, 2011.

[29] Urry J. Sociology Facing Climate Change. In: *Sociological Research Online*, Vol. 15, 2010, pp. 1-3.

[30] White KP, Ingalls RG. The Basics of Simulation. In: *Proceedings of the 2017 Winter Simulation Conference*, 2017, pp. 505-519.

[31] Wood MD, Thorne S, Butte G, Linkov I, Kovacs D. Adaptive Management for Climate Change. In: *Mental Modeling Approach, Risk, Systems and Decisions*. Springer, New York, NY, 2017, pp. 57–67.

Agent-based Simulation with Process-interaction Worldview

Shufang Xie, Tao Zhang*, Oliver Rose

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

SNE 29(4), 2019, 169 - 177, DOI: 10.11128/sne.29.tn.10492
Received: May 10, 2019 (Selected ASIM SST Hamburg 2018 Postconf. Publ.), Accepted: Sept. 10, 2019
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. After analyzing the agent-based simulation (ABS), we realize that most researchers focus on the agent-based modeling, but only a few researchers pay attention to the simulation. The reason is that the agent-based model (ABM) can run directly in a real-time manner by communication among agents and the running of the ABM is already one kind of simulation. However, the 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 introduce a process-interaction worldview (PIW) originated in the discrete event simulation to the ABS. A method for combining the agent-based simulation and the PIW is proposed. The method is validated by applying in a simple queue system and compared with the normal ABS with a timescale.

Introduction

There still exists controversy over the definition of agents. One concept of the agent appears in the distributed intelligence in Artificial Intelligence [1], in which besides distributed, autonomous, and social features, agents have to be intelligent, such as being able to perceive, learn, and adapt to the environment. Most of the researchers following this concept direct towards the multi-agent system [2], and some researchers focus on the hardware agent [3] which has been widely used in the robotics [4]. Another concept , which is derived from the emergence theory [5] in which a key notion is that simple rules generate complex behaviors, in other words, system properties emerge from its constituent agent interactions [6], is very similar to the cellular automata [7] in which agents are asked to keep simple and short, which is contrary to the first concept because the intelligence certainly makes agents more complex. However, they also have a lot in common, such as autonomy, society, distribution and so on. In most research, the contrary parts of the agents seem to be discarded. Both the intelligence and the simplicity are not given weight, but much attention is paid to the common parts, autonomy, society, and distribution [8-10]. These research lead to the formation of a new bottom-top modeling method [11], i.e., agent-based modeling, in which a system is modeled as a collection of autonomous agents. Based on a set of rules each agent individually assesses its situation, makes decisions and may execute various appropriate behaviors for the system it represents [6]. The agent-based modeling is kind of similar to the object-oriented modeling [12], but more flexible and more natural to describe the system. There are no strict requirements of intelligence or simplicity for the agents in the agent-based model.

Now we define the ABS. The ABS is the process of designing an ABM of a real system and conducting experiments with this model to understand the behavior of the system and evaluating various strategies for the operation of the system [13].

At present, the researches on the ABS mainly focus on two directions, agent-based modeling, and its application. The first one is trying to build a standard, and universal framework for the modeling [14-16]. Autonomy, society, and distribution involve in. Another one focuses on creating domain agents through studying their attributes and behaviors [17-19]. Both have provided lots of approaches and mechanisms for agent-based modeling and application in practice. However, only a few researches focus on the simulation and its efficiency.

The study will discuss the efficiency of the non-real-time ABS and combine the ABS with the discrete event simulation to boost the efficiency of the simulation. The paper is structured as follows: In Section 1, we present the efficiency problem of the ABS and analyze some worldviews which may speed up the simulation. In Section 2, a procedure for introducing process-interaction worldview to the ABS is proposed. In Section 3, some experiments are carried out to validate the procedure. The paper is concluded in Section 4.

1 Agent-based Simulation

1.1 Simulation in ABS

As we mentioned before, the current studies on the ABS focus on the ABM, and few researchers pay attention to the simulation in the ABS. Probably because the ABM can run directly in a real-time way[11], in which the running of the ABM is a simulation run, and there is no necessity to study the simulation separately. However, this type of simulation misses some important contents in the computer simulation such as the worldview in the simulation, also referred to as a simulation strategy. Therefore, many researchers studying on the computer simulation, especially the discrete event simulation, cannot help asking where the simulation is in the ABS. The so-called simulation in the ABS is a real-time simulation in which the simulation time equals the real time. However, the ABS is less efficient when being used in a non-real-time system. In some researches, the ABS is speeded up by giving a timescale under the condition of synchrony. There are two ways to achieve the synchronization: conservative algorithm and optimistic algorithm. The Conservative algorithm keeps the model running in sync exactly, but the optimistic algorithm allows asynchronous phenomenon to occur and then makes it synchronous, such as SimJade does. The SimJade[20] is a synchronization service for the JADE using an optimistic synchronization technique to manage the time in a distributed way. Because the optimistic synchronization techniques allow the asynchronous phenomenon to appear, the agents influenced by the asynchronous phenomenon have to roll back. So, lots of time is consumed by the rollback.

In addition, the real-time ABS with a timescale has two features: (1) there is no central time manager, and the agents move on according to their local time (computer clock); (2) The time spent on executing code is counted in the simulation time. In the real-time ABS without a timescale, the time for code execution is very short and can be ignored. On the other hand, in the real world, it also takes time while people make a decision. So, the time for executing the codes exists reasonably in the real-time ABS. However, in the non-real-time ABS with the time scale, the execution time is enlarged, and errors occur. And the worse thing is that the precision of simulation results will decrease as the timescale increases. We face a tradeoff between the efficiency and the precision. So how to speed up the ABS without losing any precision is a key issue.

1.2 Efficiency of the ABS

If back to the computer simulation again, we can find that the simulation has different efficiencies when different worldviews are used. There are two main types of the worldview, time-driven worldview, and event-driven worldview. In the time-driven worldview, the world progresses as the time is passing with a fixed increment (time step). Correspondingly, the world progresses as some events occur in the event-driven worldview which includes three sub worldviews: event scheduling, activity scanning, and process interaction. Introducing a suitable worldview into the ABS will be a good way to speed up the simulation. But some researchers [12, 20] argue that the simulation worldview violates the autonomy principle of the agents due to the centralization of time handling and sharing. So, most researchers did not study the ABS and the worldview together. Siebers even declares that the discrete event simulation is dead, long live the ABM [21]. However, the time handling is only in charge of simulation time which is independent and never affected by the agent. The local clocks in each agent merged into one sharing simulation clock does not intervene in the behavior of agents at all, and the agents still take action autonomously. Therefore, it is possible to introduce the worldview to the ABM.

A few researches have already focused on this field and obtained great progress, such as the entity-relationship and agent-oriented-relationship (ER\AOR) [22]. Agents, objects, events, and messages are entities in the ER\AOR; the agents and the objects are distinguished; the messages and the events are managed together to control the simulation time.

In the ER\AOR, it is natural to partition the simulation system into the environment simulator and some agent simulators. The environment simulator is responsible for advancing simulation time and managing the state of all external (or physical) objects and the external/physical state of each agent; a number of agent simulators are in charge of managing the internal (or mental) state of agents. By means of ER modeling and combination with the discrete event simulation, the ER\AOR has attracted extensive attention. However, we realize that the ER\AOR simulation is not a pure ABS because of the objects in the model. In the pure ABS, the objects which cannot be modeled as agents must belong to certain agents (attributes). In addition, because the conditional events or messages are involved, and the lifecycle of the agent is divided into many activities, the AOR simulation is the ABS with the activity- scanning worldview. As we all know, the activities-scanning worldview is not the most efficient one. It is better to choose a more efficient worldview for the ABS. Now we come to the next question: which worldview is more suitable and efficient for the ABS.

1.3 Worldview for the ABS

The event-scheduling worldview focuses on the events that instantaneously transform a system's state and schedule future events [23]. The advantage of this worldview is that periods of inactivity can be skipped over by jumping the clock from one event time to the next event time. The event-based approach is the most computationally efficient one of the three classical worldviews. The activity-scanning worldview focuses on activities and their preconditions (triggers) [23]. An activity's preconditions must be satisfied for an activity's operations. This worldview is less efficient than the event-scheduling worldview because it requires a frequent evaluation of conditions. The process-interaction worldview can be considered a combination (hybrid) of the activity-scanning worldview and the event-scheduling worldview [24]. It focuses on processes and the entities that flow through the processes and interact with resources [25]. The process-interaction worldview is more efficient than the activity-scanning worldview, but it is less efficient than the event scheduling worldview.

From the analysis above, we can see that the process-interaction worldview is the second most efficient worldview.

But besides efficiency, the choice of worldview should be made by considering other characteristics such as maintainability, modifiability, reusability, and ease of development. The process-interaction worldview is considered to be a natural way to describe models [26] and is closer to most people's mental model. In addition, the notion of "process" corresponds closely to the lifecycle of the agent and the implementation of the process-interaction worldview is very similar to the agent-based model. Moreover, if we use the event-scheduling worldview or the activity-scanning worldview, the flexibility, maintainability, and modifiability of the ABM will shrink. Therefore, we introduce the process-interaction worldview into the ABS to speed up the simulation.

2 Agent-based Simulation with the PIW

In this section, we will introduce an approach (ABS&PIW) which brings the process-interaction worldview into the agent-based simulation. First, we discuss agents in detail including its attributes, behaviors, and messages. And then some concepts in the ABS&PIW are defined. To make the approach more formal and rigorous, we formulate the approach. At last, we discuss the parallelism in the ABS&PIW.

2.1 Agents in the Agent-based Simulation

In our study, an agent has some attributes, behaviors, messages, and activation points (see Figure 1). Agents are endowed with behaviors to make independent decisions, and the messages are medium for their communications. Activation points are designed for the process interaction worldview.

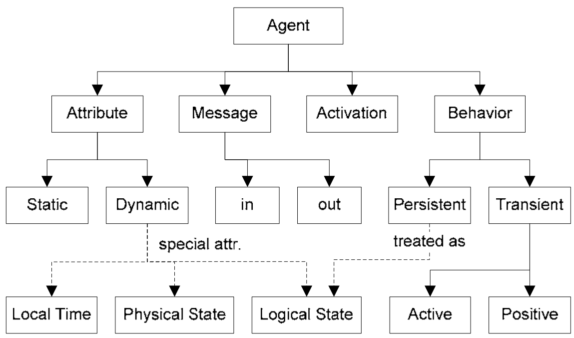

Figure 1: Attributes, behaviors, messages, and activations in an agent.

Attributes

Attributes are characteristics of the agent. An agent's attributes can be static, i.e., not changeable during the simulation, or dynamic, i.e., changeable by behaviors as the simulation progresses. For example, a static attribute is an agent's name; a dynamic attribute is an agent's memory of past interactions. The agent adapts to the environment by changing its attributes. There can be a large number of attributes in an agent, but only attributes related to the goal of the system need to be considered. The agent has three special attributes: local time, physical state, and logical state. The local time is from the inner clock in the agent, and it may be not synchronous with the simulation time. The two states will be defined in Section 2.2.

Behaviors

There are two types of behaviors: persistent behaviors and transient behaviors. The persistent behaviors are equal to activities, and they will change the state of the agent. One persistent behavior is related to one of the logical states, so the persistent behaviors are treated as logical states. Transient behaviors, which are behaviors considered in the agents, can be divided into passive behaviors and active behaviors. The passive behaviors are responsible for receiving messages and updating dynamic attributes, and the active behaviors are in charge of generating and sending messages.

Messages and Activation

Agents can receive messages from the other agents and send messages to them. The message has a given format and typically contains sender, receivers, sending time, keywords, and content. The sending time is the local time of the sender agent when the message is sent. The local time is updated with the sending time of the received messages. The concept of activation will be introduced in Section 2.2

2.2 Concepts in the ABS&PIW

Concepts in the ABS&PIW

Firstly, six concepts are given. Delays and activation points come from the process interaction method but offer some improvements.

Physical state of the agent, which has two states, active and blocked, is related to the implementation. If an agent is blocked, it gives up control of the CPU. Otherwise, it occupies the CPU.

Logical state of the agent is closely connected to the application domain and is as same as the state of the entity. The logical state is a very important dynamic attribute.

State of the agent-based model has two types, ready or unready. If the physical states of all agents are blocked, the state is ready. Otherwise, it is unready.

Straggler message is a message that its sending time is earlier than the local time of the agent who receives the message. It means a later message is received before the earlier message. The straggler message will make the simulation wrong and must be avoided. There are two ways to avoid it: a conservative algorithm which does not allow the straggler message and an optimistic algorithm which allows it and then corrects it.

Delay is a period in which the logical state of the agent is not changeable. When a delay occurs, the agent will create the next activation point and become blocked.

Activation point is the time position where the agent will be after a delay ends. The agent is activated at this point and performs actions until a new delay occurs. An activation point has such a given format including activation time, activation agent, and keywords. There are two types of activation point, conditional and non-conditional. Non-conditional activation point is explicit. In contrast, the conditional activation point is uncertain in which delays of the agents does not end until the agents meet the given condition.

Relation among some Concepts

The relation among these concepts is shown in Figure 2. The agent is similar to the active entity; the life cycle of the agent is the process of the entity and is made up of a series of activation-delay-activation. The activation point is located at the time an event occurs, and the physical state is active at this point. The agent responds to the event by a transient behavior. During a delay, an activity is carrying out, and the physical state of the agent is blocked. The activity is a persistent behavior corresponding to the certain logical state.

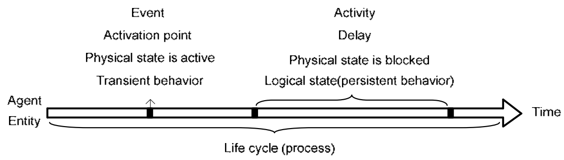

Figure 2: Relation between activation point, event, and so on.

2.3 Formulation of the ABS&PIW

Symbol Definition

t, t_{plan} current and planned simulation time

t_{sa} sampling interval time

s_{mo} state of the ABM(0-ready,1-unready)

u flag of model update (0-no update,1-update)

R_{FAL} , R^0_{FAL} activations in the future activation list and initial value

R_{CoAL} , R^0_{CoAL} activations in CoAL and initial value

R_{CuAL} current activations in CuAL

a an agent, $a \in A$,A is a set of all agents

r_a current activation point of the agent a

c_{r_a} type of activation point(1-conditional,0-uncondt.)

g_{r_a} flag of the condition(0-unmeet,1-meet)

R_a a set of activation types from the agent a

m_a a message received by the agent a

M_a^{out} a set of message types sent by the agent a

M_a^{in} a set of message types received by a

τ_{m_a} timestamp of the message m

τ_r activation time (timestamp of the activation point)

τ_a local time of agent a updated by the time stamp τ_{m_a} or τ_r

s_a physical state of the agent a

f_a agent attribute, $f_a \in F_a$

F_a , F_a^0 a set of all attributes and initial value

b agent behavior

B_a^{po} , B_a^{ac} a set of passive and active behaviors

$H_a^1 = RS(r_a, b_a^{po}) = \{h^1_{r,b} \mid r \in R_a, b \in B_a^{po}\}$ relationship between activations and passive behaviors . If b is related to r, $h^1_{r,b} = 1$, otherwise $h^1_{r,b} = 0$. The following relationships have the same rule.

$H_a^2 = RS(m_a^{in}, b_a^{po}) = \{h^2_{m,b} \mid m \in M_a^{in}, b \in B_a^{po}\}$

$H_a^3 = RS(r_a, b_a^{ac}) = \{h^3_{r,b} \mid r \in R_a, b \in B_a^{ac}\}$

$H_a^4 = RS(m_a^{in}, b_a^{ac}) = \{h^4_{m,b} \mid m \in M_a^{in}, b \in B_a^{ac}\}$

$H_a^5 = RS(b_a^{po}, f_a) = \{h^5_{b,f} \mid b \in B_a^{po}, f \in F_a\}$

$H_a^6 = RS(b_a^{ac}, a', m_a^{out}) = \{h^6_{b,a',m} \mid, b \in B_a^{ac}, a' \in A, m \in M_a^{out}\}$

$H_a^7 = RS(b_a^{ac}, r_a) = \{h^7_{b,r} \mid, b \in B_a^{ac}, r \in R_a\}$

Four-tuple of the ABS&PIW

We provide a mathematical framework for the ABS following the process interaction worldview. The simulation is specified as a four-tuple

$$SIM = (I, TM, ABM, O) .$$

I is a set of inputs

$$I = (R^0_{FAL}, R^0_{CoAL}, \{F_a^0 \mid a \in A\})$$

including initial activation points and initial attributes of all agents. O is a set of outputs

$$O = (\{(t, F_a^t) \mid a \in A, 0 < t < t_{aim}, \ \mathrm{mod}(t, t_{sa}) = 0\}) ,$$

which is made up by attributes of all agents at each sample point. TM is a time manager

$$TM = (t, t_{plan}, \ R_{FAL}, R_{CoAL}, R_{CuAL})$$

who is in charge of the simulation time and manages all of the activation points created by agents. The activation points are grouped into three lists: conditional activation list (CoAL), future activation list (FAL) and current activation list (CuAL). FAL and CoAL are direct lists in which the activation points come from agents. CuAL is an indirect list in which the activation points are moved in from FAL. The planned time t_{plan} is the maximal simulation time. The simulation will end when the simulation time t reaches the planned time. Note that there is no TM in the general real-time ABS. ABM is an agent-based model described as

$$ABM = (s_{mo}, u, \{(\tau_a, s_a, F_a, B_a^{po}, B_a^{ac}, M_a^{in}, M_a^{out}, R_a, RS_a) \mid a \in A\}) .$$

The model state s_{mo} and the updated flag u are used by the *TM* to advance the simulation. The simulation clock advances whenever the model state is ready and the model does not update anymore. In this way, we can avoid the straggler messages and ensure that all conditional activation points which meet the corresponding conditions are activated as soon as possible. This is a conservative synchronization algorithm. The ABM is different from the general one which contains three elements, agents, relationship, and messages [11]. In a complex system, the relationship changes dynamically and there are massive situations. It is difficult to express the relationship among all agents by a two-axis matrix. But for the individual agent, the situations of relationship with other agents are countable.

So, in our ABM, the relationship and messages are specified in the individual agents and the target agents with corresponding messages can be got by some simple IF-THEN rules. RS_a is a set of the relationship sets, $RS_a = (H_a^1, H_a^2, H_a^3, H_a^4, H_a^5, H_a^6, H_a^7)$, which contains seven relationships in the agent such as the relationship between received messages (in) and passive behaviors, as well as the relationship between active behaviors, target agents and corresponding messages (out). Local time and physical state are two special attributes and play a great role in the simulation. We extract them from the attributes and consider separately. The physical state is used to determine the model state.

Procedure of the ABS&PIW

The following is a procedure for the simulation. Simulation initialization (1), advancing time (2), and activating agents (3) are executed by the *TM*. The simulation is initialized with the inputs. Simulation clock t advances according to the time of the earliest activation points. All concurrent activation points, including both current activation points and conditional activation points, are activated at a time.

(1) Initialize

$t = 0$, $R_{FAL} = R_{FAL}^0$, $R_{CoAL} = R_{CoAL}^0$

$F_a = F_a^0$, $s_a = 0$, *where* $a \in A$

$s_{mo} = 1$

(2) Advance time

if $R_{FAL} = \phi$ *or* $t > t_{plan}$ *then* ***simulation ends***

$\tau_{\min} = \min(\{\tau_r \mid r \in R_{FAL}\})$

$R_{CuAL} = \{r \mid \tau_r = \tau_{\min}, r \in R_{FAL}\}$

$R_{FAL} = R_{FAL} - R_{CuAL}$

$t = \tau_{\min}$

(3) Activate

$R' = \{R_{CoAL}, R_{CuAL}\}$, $R_{CuAL} = \phi$

if $R' = \phi$ *then go to (2)*

$A' = \{a' \mid r_{a'} \in R'\}$

$S_{A'} = 1$ *,where* $S_{A'} = \{s_{a'} \mid a' \in A'\}$

TM activates A' *at* R'

(4) When an agent $a' \in A'$ ***is activated at*** $r_{a'} \in R'$

$\tau_{a'} = \tau_{r_{a'}}$

Get passive behavior b_1 *by* $H_{a'}^1$,

which satisfies $h_{r_{a'},b_1}^1 = 1, h_{r_{a'},b_1}^1 \in H_{a'}^1$

if $c_{r_{a'}} = 1$ *then*

$g_{r_{a'}} = g_{r_{a'}}^{new}$ *,where* $g_{r_{a'}}^{new}$ *is new one from b1*

If $g_{r_{a'}} = 0$ then $s_{a'} = 0$ *and go to (6)*

end if

if $F_{a'}' \neq \phi$ *then* $F_{a'}' = F_{a'}^{n'}$,

where $F_{a'}^{n'}$ *are new values calculated by* b_1

and $F_{a'}' = \{f \mid h_{b_1,f}^5 = 1, f \in F_{a'}, h_{b_1,f}^5 \in H_{a'}^5\}$

u=1

Get active behavior b_2 *by* $H_{a'}^3$

which satisfies $h_{r_{a'},b_2}^3 = 1, h_{r_{a'},b_2}^3 \in H_{a'}^3$

if $\exists r_{a'}' \in R_{a'} : h_{b_2,r_a'}^7 = 1, h_{b_2,r_a'}^7 \in H_{a'}^7$ *then*

if $c_{r_{a'}'} = 0$ *then* $R_{FAL} = R_{FAL} + \{r_{a'}'\}$

if n $c_{r_{a'}'} = 1$ *then* $R_{CoAL} = R_{CoAL} + \{r_{a'}'\}$

end if

$(A'', M_{a'}^{out'}) = \{(a'', m) \mid h_{b_2,a'',m}^6 = 1, h_{b_2,a'',m}^6 \in H_{a'}^6\}$ *if* $A'' \neq \phi$ *then*

$S_{A''} = 1$ *,where* $S_{A''} = \{s_{a''} \mid a'' \in A''\}$

send messages $M_{a'}^{out'}$ *to* A''

end if

if $c_{r_{a'}} = 1$ *then* $R_{CoAL} = R_{CoAL} - \{r_{a'}\}$

$s_{a'} = 0$

go to (6)

(5) When an agent $a'' \in A''$ ***receives the message*** $m_{a''}$ ***from the agent*** a'

It is similar to step (4) and just needs the following replacements and to ignore condition activation:

$a' \to a''$, $r_{a'} \to m_{a''}$, $H_{a'}^1 \to H_{a''}^2$, $H_{a'}^3 \to H_{a''}^4$

(6) When an agent is blocked

if $\forall a \in A : s_a = 0$ *then* $s_{mo} = 0$

if $s_{mo} = 0$ *then*

if $u = 1$ *then* $u = 0$ *and go to (3)*

if $u = 0$ *then go to (2)*

end if

In steps (4, 5), when an agent becomes active or receives messages, passive behaviors handle the received messages or activations and update its attributes. Active behaviors create new activation points and communicate with others. Decisions on the timing of advancing the time and quitting repeat of the conditional activations (6) are made by ABM according to the model state and the updated flag whenever the physical state of one agent becomes blocked.

Parallelism in the ABS&PIW

In the agent-based simulation, agents run in parallel. While concurrent activation points are activated simultaneously, associated agents will respond in parallel. The parallelism in ABS is shown in Figure 3. To avoid the straggler messages mentioned above, we adopt the conservative synchronization algorithms to ensure the correct local time (see step 6). Even though it cannot fully take advantage of parallelism, it can prevent the straggler messages from appearing at all and save the rollback time in the optimistic algorithm.

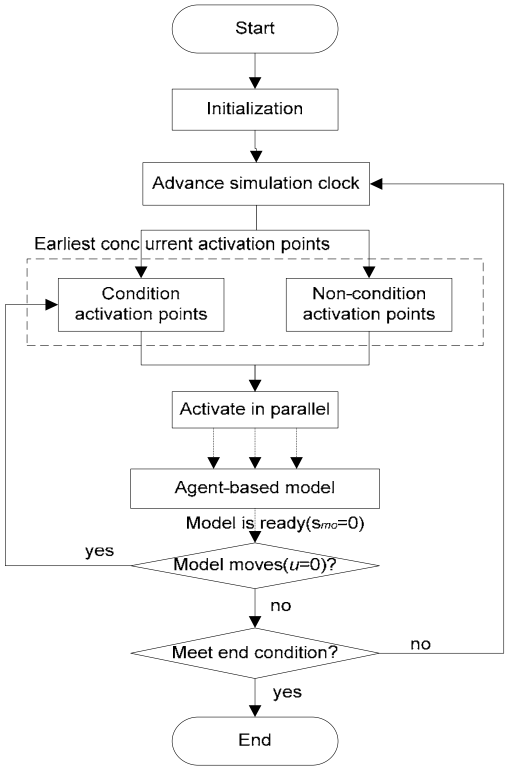

Figure 3: Parallelism in the agent-based model.

3 Experiments

A queuing system $M/M^r/1$ with a batch service is one of the classical discrete event systems. We use it to validate the proposed approach. An ABM of the system is built for the queue system, and the simulation result is compared with the theoretical value. We also compare the efficiency of the approach with the real-time ABS by using the built model.

3.1 Queuing System $M/M^r/1$

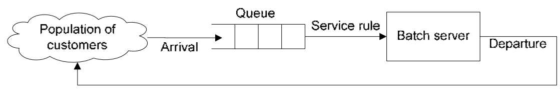

Figure 4: A Queuing system.

The queuing system $M/M^r/1$, shown in Figure 4, consists of an infinite population of customers, an infinite queue with FCFS (First Come First Serve) dispatching rule, and one batch server. The batch server provides service in batches (of size r) for arrived customers based on the rule. Customers who arrive and find the server busy join in the queue. Customers in a batch start service at the same time and depart together after served. The interval arrival and service times follow exponential distributions.

3.2 Agent-based Model of the $M/M^r/1$ Queuing System

Three types of agent are abstracted from the queuing system: customer source, customer, and server. Because a server is also an active entity, activation points of the whole system are simplified to two types: customer arrival and service completion. We build a single group ABM for the queuing system. The customer source generates customers according to a certain time distribution. The behaviors of the customers are requesting service, joining the queue to wait, accepting service, and leaving the system. The behaviors of the server include handling customer messages and providing service. The queue is part of the server agent. After a customer is served, the queue will use the given rule to choose new customers to begin service.

3.3 Correctness Verification

Assuming that customers arrive one by one; arrival rate λ is 9.76 per hour; the service batch r is 3; service rate μ is 5 per hour. Measures of performance in the steady state can be calculated with the following formulas [27].

The probability that n customers are in the system,

$$p_n = \begin{cases} (1-s_0^{-n-1})/r & 0 \le n < r \\ \rho(s_0-1)s_0^{r-n-1} & r \le n \end{cases}$$

where $\rho = \lambda/(r\mu)$, s0 satisfies $|s_0| > 1$ and

$$r\rho s_0^{r+1} - (1+r\rho)s_0^r + 1 = 0$$

The average number of customers in the system,

$$\overline{N} = (r-1)/2 + 1/(s_0-1)$$

The average number of customers in the queue,

$\overline{N}' = (r-1)/2 + 1/(s_0 - 1) - r\rho$

The average waiting time of customers,

$\overline{W} = (r-1)/(2\lambda) + 1/(r\mu(s_0 - 1))$

The average time of customers in the system,

$\overline{T} = (r-1)/(2\lambda) + 1/(r\mu(s_0 - 1)) + 1/\mu$

Results	p_0	$p(n>0)$	$\overline{N}'$	$\overline{N}$	$\overline{W}$ (min)	$\overline{T}$ (min)
Theoretical	0.067	0.933	3.05	5.00	0.37	0.57
ABS&PIW	0.067	0.933	3.01	4.96	0.31	0.51

Table 1: Comparison between theoretical result and simulation result.

3.4 Comparison with the Real-time ABS (Time Scale)

In order to prove the less efficiency and precision, a brief simulator for the real-time ABS with timescale is developed by using JADE (Java Agent Development Environment). In JADE, the messages are not in sync sent and received (asynchronous communication). To achieve the simulation with the timescale, we improved it to the synchronous communication. After improvements, an agent (1) who just sends a message (a) will move on only after the message (a) is received and handled by its receiver agent (2). If the receiver (2) needs to send another message (b) in the message (a)'s handling process, the agent (1) has to wait for the receiver (2) until its message (b) is received and handled by another receiver. In addition, when a delay occurs in the agent, the agent will be blocked until the delay ends. The timescale is used in such delays to decrease the delay time so as to speed up the simulation.

We still use the queuing system but with the constant arrival rate (3 per hour) and service rate (3 per hour) to avoid the stochastic influences. The theoretical values can be got easily, shown in Table 2. The simulation runs ten days, and the simulation results from our approach and the real-time ABS are shown in Table 2 too.

We can see that the results of our approach are as same as the theoretical values. It takes only 0.374 seconds (the configuration of hardware is Intel i3-330M 2.13GHz CPU and 2GB memory). However, for the ABS with the timescale, the error is very big, and it also took the longer time.

4 Conclusions

Because the PIW is more natural and closer to the mental model, it is combined with the ABS to speed up the simulation. The ABS&PIW approach is proposed on the basis of the agent-based model. We provide 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. A conservative algorithm is adopted to avoid straggler messages while the simulation is running. The result from an application to the queuing system $M/M^r/1$ shows the validity of the proposed approach. Comparing the efficiency with the real-time ABS, it performs more efficiently. Besides the advantages mentioned above, the ABM can be naturally combined with the process interaction worldview. The flexibility, maintainability, and modifiability of the ABM are also enhanced in this way.

Results	p_0	$p(n>0)$	$\overline{N}'$	$\overline{N}$	$\overline{W}$ **(min)**	$\overline{T}$ **(min)**	**Time spent (s)**
Theoretical	0.000	1.000	1.00	4.00	20.0	80.0	-
ABS&PIW	0.000	1.000	1.00	4.00	20.0	80.0	0.374
ABS (scale 70000)	0.002	0.998	1.07	3.20	24.6	86.4	12.300
ABS (scale 500000)	0.041	0.959	1.05	1.85	147.5	300.4	1.743

Table 2: Comparisons with real-time ABS (time scale).

References

[1] Sugumaran V. Distributed Artificial Intelligence, Agent Technology, And Collaborative Applications. 2008: Information Science Reference.

[2] Lian J, Shatz SM, He X. Flexible coordinator design for modeling resource sharing in multi-agent systems. Journal of Systems and Software, 2009. 82(10): p. 1709-1729.

[3] Tapia DI, et al. Integrating hardware agents into an enhanced multi-agent architecture for Ambient Intelligence systems. Information Sciences, (0).

[4] García C, et al. Design and modeling of the multi-agent robotic system: SMART. Robotics and Autonomous Systems, 2012. 60(2): p. 143-153.

[5] Namatame A. Collective Intelligence of Networked Agents, in Emergent Intelligence of Networked Agents, A. Namatame, S. Kurihara, and H. Nakashima, Editors. 2007, Springer Berlin Heidelberg. p. 159-176.

[6] Bonabeau E. Agent-based modeling: Methods and techniques for simulating human systems. Proceedings of the National Academy of Sciences of the United States of America, 2002. 99(Suppl 3): p. 7280-7287.

[7] Wolfram S. Cellular automata and complexity. 1994: Addison-Wesley Pub. Co.

[8] Váncza J, Márkus A. An agent model for incentive-based production scheduling. Computers in Industry, 2000. 43(2): p. 173-187.

[9] Oliva G, Panzieri S, Setola R. Agent-based input–output interdependency model. International Journal of Critical Infrastructure Protection, 2010. 3(2): p. 76-82.

[10] McLane AJ, et al. The role of agent-based models in wildlife ecology and management. Ecological Modelling, 2011. 222(8): p. 1544-1556.

[11] Macal CM, North MJ. Tutorial on agent-based modelling and simulation. J of Sim, 2010. 4(3): p. 151-162.

[12] Davidsson P. Multi Agent Based Simulation: Beyond Social Simulation, in Proceedings of the Second International Workshop on Multi-Agent-Based Simulation-Revised and Additional Papers. 2001, Springer-Verlag. p. 97-107.

[13] Shannon RE. Systems simulation: the art and science. 1975: Prentice-Hall.

[14] Botta N, et al. A functional framework for agent-based models of exchange. Applied Mathematics and Computation, 2011. 218(8): p. 4025-4040.

[15] Fortino G, Russo W. ELDAMeth: An agent-oriented methodology for simulation-based prototyping of distributed agent systems. Information and Software Technology, 2012. 54(6): p. 608-624.

[16] Grimm V, Berger U, Bastiansen F. A standard protocol for describing individual-based and agent-based models. Ecological Modelling, 2006. 198(1–2): p. 115-126.

[17] Chen L. Agent-based modeling in urban and architectural research: A brief literature review. Frontiers of Architectural Research, 2012. 1(2): p. 166-177.

[18] Rebaudo F, Dangles O. An agent-based modeling framework for integrated pest management dissemination programs. Environmental Modelling & Software, (0).

[19] Cao Y, Chen X-h. An agent-based simulation model of enterprises financial distress for the enterprise of different life cycle stage. Simulation Modelling Practice and Theory, 2012. 20(1): p. 70-88.

[20] Pawlaszczyk D, Timm IJ. A hybrid time management approach to agent-based simulation, in Proceedings of the 29th annual German conference on Artificial intelligence. 2007, Springer-Verlag: Bremen, Germany. p. 374-388.

[21] Siebers PO, et al. Discrete-event simulation is dead, long live agent-based simulation! J of Sim, 2010. 4(3): p. 204-210.

[22] Wagner G. AOR Modelling and Simulation: Towards a General Architecture for Agent-Based Discrete Event Simulation, in Agent-Oriented Information Systems, P. Giorgini, B. Henderson-Sellers, and M. Winikoff, Editors. 2004, Springer Berlin / Heidelberg. p. 174-188.

[23] Miller JA, et al. Investigating Ontologies for Simulation Modeling, in Proceedings of the 37th annual symposium on Simulation. 2004, IEEE Computer Society. p. 55.

[24] Zeigler B, Praehofer H, Kim T. Theory of Modeling and Simulation, Second Edition. 2000: Academic Press.

[25] Banks J, Carson JS. Process-interaction simulation languages. SIMULATION, 1985. 44(5): p. 225-234.

[26] Franta WR, Maly K. An efficient data structure for the simulation event set. Commun. ACM, 1977. 20(8): p. 596-602.

[27] Gelenbe E, Pujolle G. Introduction to queueing networks. b, ed. a. Vol. c. 1987, New York: John Wiley &Sons. 900.

Simulation-based Evaluation of Dynamic Vehicle Routing Problem Features for Algorithm Selection

Thomas Mayer*, Tobias Uhlig, Oliver Rose

Institute of Computer Engineering, Universität der Bundeswehr München, Werner-Heisenberg-Weg 39, 85579 Neubiberg, Germany; *thomas.mayer@unibw.de

SNE 29(4), 2019, 179-188, DOI: 10.11128/sne.29.tn.10493
Received: May 10, 2019 (Selected ASIM SST Hamburg 2018 Postconf. Publ.), Accepted: Sept. 10, 2019
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Algorithm selection based on problem instance features is a common method to choose approaches for NP-hard combinatorial optimization problems. Features concerning the location of nodes for the Vehicle Routing Problem (VRP) are intensively studied. Although the dynamic version of the problem is gaining steadily in importance, there is a lack in research targeting problem features describing the dynamism of an instance. In our paper we translate known VRP features to features for the Dynamic Vehicle Routing Problem (DVRP). We investigate the suitability of them for algorithm selection. To this end, we explore the performance space of a Greedy and a Re-planning algorithm for various dynamic problem instances using simulation and an evolutionary algorithm. For the algorithm selection we investigate our features in combination with a supervised machine learning technique. The applicability of our features is demonstrated in a use case for autonomous algorithm selection of DVRP instances.

Introduction

In the last years, the research interest in vehicle routing has significantly increased due to our modern way of life. We order different goods of daily use over the Internet, we share cars and bicycles and use on-demand transportation services. To provide all these services, complex Vehicle Routing Problems (VRPs) with several constraints have to be managed and solved. The fundamental VRP was introduced by Dantzig and Ramser in [1] as generalization of the Traveling Salesman Problem (TSP) introduced from Flood in [2]. Essentially, it encompasses the planning of routes for vehicles to satisfy customer demands. The routes usually start and end at a central depot. A current survey and taxonomy on the VRP is, for example, [3]. Especially, the advances of information and communication technologies and the changing nature of the real-world problem instances put the dynamic version of the VRP into the research focus. Pillac et al. in [4] and Psaraftis et al. in [5] describe an explosion in the number of related papers after the year 2000. The Dynamic Vehicle Routing Problem (DVRP) was first introduced from Wilson and Colvin in [6], with a description of a computer controlled Dial-A-Ride system in Rochester, NY (USA). The DVRP is an extension of the VRP and is characterised by changing problem parameters over the time, [4], for example, additional customer demands occur during route execution. A current survey on the DVRP is provided in [7]. As a result of the increased research interest, an increasing number of different solution approaches to tackle the DVRP are available. Psaraftis et al. identifies the five main solution methods: *Tabu Search*, *Various Neighborhood Search*, *Insertion Methods*, *Dynamic Programming*, and *Markov Decision Process* in [5]. For each method several different heuristics and implementations exist. Simulation is also commonly used to handle the dynamic and stochastic aspects of vehicle routing. For example Juan et al. improves the Clarke and Wright heuristic, [8], by Monte Carlo Simulation, [9]. Juan et al. provide a review of methods using simulation to solve combinatorial optimization problems in [10]. The concept of Simheuristic is introduced, a methodology that integrates simulation into the solution finding process for combinatorial optimisation problems. [11] use this methodology to solve the Two-Echelon Location Routing Problem, a combination of the Capacitated Location Routing Problem (CLRP) and a VRP. Due to the intense research in this area and the increased number of algorithms and solution approaches for different variations of the

DVRP, the following research questions arise: Which is the best solution approach or algorithm for a specific problem instance? And, is it possible to distinguish between problem instances to select the algorithm that has a better expected performance? With regard to the No Free Lunch Theorem introduced in [12], we know that there is no strictly superior heuristic that outperforms all other heuristics for all problem instances. That means some problem instances are more difficult or harder to solve for certain algorithms compare to others. Our research focuses on learning which problem instances are easier or harder for a certain algorithm based on problem features. The acquired knowledge can be used to select the likely better performing algorithm, means the algorithm that likely delivers the best solution. The question for the best algorithm given a problem instance is not only arising in the context of vehicle routing. It is a general problem, formalised and introduced by Rice in [13] as Algorithm Selection Problem (ASP). The bases for algorithm selection are features characterising the problem instances. For the static version of the VRP several problem features, considering the location of the customer demands, exist, [14]. But features for the dynamic version of the problem are rarely available, [15]. Our work closes this gap by transferring existing location-based TSP and VRP features, which we discuss in Section 1, into dynamic problem features outlined in Section 2. To validate the suitability of our features for algorithm selection we use them for algorithm performance prediction and autonomous algorithm selection supported by simulation. To this end, we first evolve dynamic problem instances from existing static instances with the help of the general purpose metaheuristic optimisation framework SEREIN introduced in [16]. The evolved problem instances are solved with a Greedy and a Re-planning algorithm, which we implemented in the open-source Rich Vehicle Routing Problem Simulator (RVRP Simulator) introduced in [17]. With the generated dynamic instances and the simulation results we are able to investigate our introduced features and whether they are suitable to meaningful distinguish between problem instances. The results of this investigation and the dynamic instance generation are also outlined in Section 2. To validate our problem features in the context of algorithm selection, we implemented a machine learning technique to model the relationship between the problem features and the algorithm performance. The basis for the implemented supervised learning approach is data which were collected with the help of simulation. This model is validated with the grouped cross validation method and additionally evaluated by algorithm performance prediction for unseen problem instances. The model and its evaluation are outlined and discussed in Section 3. Our introduced DVRP features show their validity as base for autonomous algorithms selection. Finally, conclusions are drawn in Section 4.

1 Related Work

This Section introduces the DVRP and the ASP and outlines current research about algorithm selection in the domain of vehicle routing. Additionally common location-based features, used in algorithm selection and performance evaluation studies for the TSP and the VRP, are discussed.

1.1 The Dynamic Vehicle Routing Problem

The DVRP was first introduced in [6] as an extension of the VRP, with a description of a computer-controlled Dial-A-Ride system in Rochester, NY (USA). The VRP, a generalization of the TSP, is defined on a graph $G = (V,E,C)$, where $V = \{v_0,\dots,v_n\}$ is a set of nodes. The set $E = \{(v_i,v_j)|(v_i,v_j) \in V^2; i \neq j\}$ defines the neighborhood of the nodes V as edge set. $C = (c_{ij})_{(v_i,v_j)} \in E$ defines a cost matrix over E. Nodes are usually customer demands, characterized by a location, which needs to be serviced. The DVRP extends the VRP notion by adding a point in time $(t_i)_{v_i}$ for all v_i, that defines when the customer request appears. In a VRP instance all $(t_i)_{v_i}$ for all v_i would be 0. In a dynamic setting (DVRP) at least one $(t_i)_{v_i}$ would be greater than 0. The TSP, VRP, and DVRP have in common, that the main characteristics of a node $v_i \in V$ is the location.

1.2 The Algorithm Selection Problem

The ASP is formalised and introduced by Rice in [13]. Figure 1 visualises the framework for the ASP proposed in [13]. It aims to predict the best algorithm performance for a problem instance, based on measurable features, [18]. The following four main components are part of the framework.

- the problem space P contains a set of problem instances;
- the feature space F represents a set of measurable features, which can be extracted from all problem

instance in P; the feature extraction process has to be less time-consuming compared to solving the problem instances with algorithms from the algorithm space A;

- the algorithm space A contains a portfolio of algorithms which are able to solve the problem instances from P;
- the performance space Y describes a mapping from algorithm result to a performance metric.

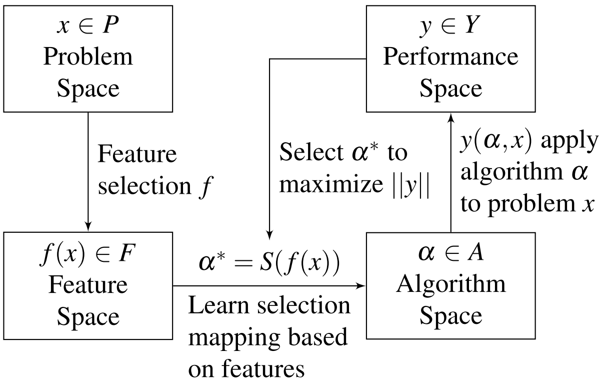

Figure 1: A framework for the Algorithm Selection Problem (ASP) introduced from Rice in [13].

The main challenge of the ASP is to find a selection mapping $S(f(x))$ into the algorithm space A for a problem instance $x \in P$ with feature vector $f(x)$. The mapping has to maximise the performance metric $||y||$ for $y(\alpha, x)$, where α is an algorithm in A. A broad discussion on algorithm selection across a variety of disciplines is given in the survey paper [19].

In the domain of static vehicle routing, [20] investigates the TSP difficulty by learning from evolved instances. The work uses an evolutionary algorithm to evolve TSP instances which are intentionally easy or hard for algorithms based on the Lin-Kernighan heuristic method [21]. Features are derived from the problem instances and the impact of these features for the difficulty of the algorithms is investigated. [22] investigates the success of 2-opt based local search algorithms for solving the TSP and shows important features that make problem instances hard or easy for 2-opt approaches. [23] studies algorithm selection for the Traveling Thief Problem (TTP) based on TSP features. The TTP is a combination of the TSP and the Knapsack Problem (KP). Earlier work focuses on the impact of problem features for the problem difficulty in general. For example, [24] demonstrates that the variance of the distance matrix correlates with the TSP difficulty for exact solving approaches. [25] shows that this is also true for heuristic solving approaches.

1.3 Problem Features

The algorithm selection and performance evaluation studies in the domain of static vehicle routing are usually based on features involving the location of the customer demands. Beside the number of customer demands, [14] classifies the 47 available VRP features, used in their study, into the following eight groups.

- *Distance Features* summarise features which involve the costs of the customer demands. The cost of a demand is the distance from the location of the customer to the depot. Considered are the lowest, highest, mean and median costs of all customer demands. Additionally statistics about the distance matrix, like distinct distances, or the standard deviation of the matrix are categorised in this group.
- *Mode Features* group all features concerning the mode of the customer demand cost distribution, [26]. For all customer the costs to all other customers are calculated. The mode is the cost value which appears most often. Additionally the frequency, quantity and mean of the mode values are considered. [14] additionally groups the number of modes of the customer demand cost distribution introduced in [27] in this category.
- *Cluster Features* bundle features based on statistically information extracted from the results of a clustering algorithm, like the number of clusters or the mean distance of the customer demands to the cluster centroid.
- *Nearest Neighbour Distance Features* group features derived from the calculated nearest-neighbour distances for each customer demand, like the minimum, maximum, mean, or median.
- *Centroid Features* summarise all features calculated dependent on the centroid of all customer demands, including the coordinates of the centroid, minimum, maximum, mean and median of the distances between customer demands and centroid.
- *MST Features* bundle features based on characteristics of the calculated Minimum Spanning Tree

(MST). Considered are the minimum, mean, median, maximum and standard deviation of the depth and distance of the MST.

- *Angle Features* bundle features based on the calculated angle between the two nearest neighbour customer demands, like the minimum, mean, median, maximum and standard deviation of all calculated angles.

- *Convex Hull Features* group the features concerning the convex hull from the problem instance, like the area of the hull and the fraction of customer demands which are part of the hull.

Problem features for the DVRP are rarely described in the literature and are usually discussed independently from algorithm selection. [28] introduced the Degree of Dynamism (DOD), a measure of the dynamism of a dynamic routing problem. The DOD describes the ratio between dynamic customer demands and the sum of dynamic and static customer demands. The DOD is similar to the feature *Number of customer demands* in the context of static vehicle routing. [29] developed a framework based on the DOD. The framework classifies weak, moderate and strong dynamic systems and recommends solution approaches for these classes of dynamic routing problems. [29] also introduced the Effective Degree of Dynamism (EDOD) as extension of the DOD. The EDOD considers the planning horizon T for the calculation of the measure of the dynamism of a routing problem. [29] also studies the relation between DOD, EDOD and routing costs for the Partially Dynamic Traveling Repairman Problem (PDTRP). In [15] we introduced the Location-based Degree of Dynamism (LDOD) capturing the location of the dynamic customer request. We show that there is a positive correlation between our proposed LDOD and the resulting DVRP solution quality (performance) for a Greedy and a Re-planning algorithm. The LDOD is the only feature that focuses on the location of the dynamic customer requests. We categorise the LDOD in the group of *Distance Features*. The LDOD is the only DVRP instance feature that is based on the fundamental characteristic location of a node (customer request). A fundamental problem characteristic is an inherent property, i.e., it is applicable regardless of the problem variation, compare [15].

2 New Features for the Dynamic Vehicle Routing Problem

In this Section DVRP features derived from static VRP location-based features, which are building the Feature Space F, are presented and evaluated. The generation of the Problem Space P, Performance Space Y and Algorithm Space A are also discussed within this Section.

2.1 Feature Space

Since problem features for the DVRP are rarely described in the literature, compare Section 1.3, we had to introduce new problem features. The TSP, the VRP and the DVRP are all defined on a graph G containing a set of nodes $V = \{v_0, \ldots, v_n\}$, compare Subsection 1.1. The main characteristic of a node v_i is its location. Based on this consideration, we are able to base our location-based features for the DVRP on location-based features introduced for the TSP. In general there are two main groups of features we consider to characterise a DVRP problem instance. In the first group we classify existing TSP features which we applied to the dynamic customer demands only. Here we consider *Angle Features*, *MST Features*, *Nearest Neighbour Distance Features*, *Distance Features*, and *Cluster Features*. Features from the second group try to capture the dynamism of the problem instance, by considering both, dynamic and static customer demands. Known examples for the second group are the DOD ([28]) and the LDOD ([15]). For our research we transferred *Angle Features*, and *Nearest Neighbour Distance Features* discussed in Section 2 into features considering dynamic and static demands. Therefore we calculated the distances and angles between dynamic and nearest neighbour static customer demands, compare Figure 2. Since a route to a dynamic customer demand will likely not start at the nearest neighbour static customer, we considered neighbourhoods with size of 2, 3, and 5. For all neighbourhoods we determined features considering the sum, mean, median and maximum value. For example for the neighbourhood of 2 we calculated the distances a and b, compare Figure 2. For the distances we calculated the values $sum(a,b)$, $mean(a,b)$, $median(a,b)$, and $max(a,b)$ for all dynamic customer demands. The resulting features are the sum, maximum, mean, median, standard deviation, and variance coefficient of these calculated values.

We also considered *Centroid Features* and transferred them into features considering dynamic and

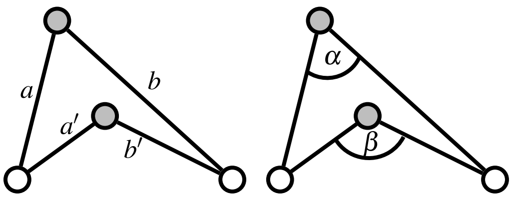

Figure 2: Distances and angles between dynamic customer demands (grey coloured circles) and nearest neighbour static customer demands (white coloured circles).

static customer demands. To this end, we calculated the centroid of the instance including both dynamic and static customer demands. We determined the average and the sum distance from dynamic and the average distance from all static customer demands to the centroid. The resulting feature is the ratio between these two distances. The last feature we introduced is considering the ratio between the surface areas spanned by all dynamic and all static customer demands. Overall we determined 184 different DVRP features.

2.2 Problem Space

To investigate the suitability of our introduced features for algorithm selection we had to create a problem space P and a corresponding performance space Y with algorithms $\alpha \in A$, compare Figure 1. We investigated the suitability of our introduced features with the most basic version a DVRP. We are not considering any capacity constraint or any restrictions regarding the customer requests like time windows or service times. We are focusing on problem instances with one depot and one vehicle only. To expand the problem space as wide as possible, we applied the general purpose metaheuristic optimisation framework SEREIN introduced in [16] to derive dynamic from static VRP instances. Figure 3 visualises the four static VRP instances from [30], which we considered for our research. They have either uniformly distributed (CMT01, CMT04) or clustered (CMT11, CMT12) customer requests.

By deriving dynamic from static problem instances three decisions have to be made. First, how many customer requests shall be dynamic? Second, which requests exactly shall be dynamic? Third, when the customer demand appears during the time horizon? The number of dynamic customer demands is determined by the DOD. In our research the following $DOD \in \{0.1, 0.2, \ldots, 0.9\}$ are considered. The second

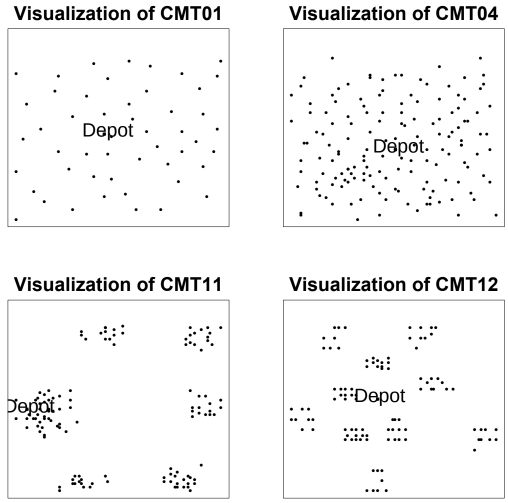

Figure 3: Subset of static VRP instances introduced in [30], considering equal distributed and clustered customer demands (black dots).

decision is made by SEREIN. It selects the determined number of dynamic requests out of all available request. The genetic algorithm evaluates the performance (solution costs) of the created instance with the help of the RVRP Simulator introduced in [17]. The RVRP Simulator provides a set of standard algorithms to handle the dynamic requests. For the problem instance generation we choose a simple Greedy algorithm provided in the simulator. For a more detailed description of the implemented algorithm see the following Section. The solution costs are used to navigate through the search space. The main task of SEREIN is to create dynamic problem instances where the Greedy algorithm performs very good (low solution costs, easy problem instances for the Greedy algorithm) and also instances where the algorithm performs very bad (high solution costs, hard problem instances for the Greedy algorithm). The third decision, when the dynamic requests appear, is not made by the genetic algorithm. Currently, we intend to minimise the influence of the time t_i when request i appears. It is obvious that requests appearing very late are probably producing higher routing costs. For example, if all dynamic requests appear when the vehicle finished its initial route planned for the static customer requests, the employed algorithm cannot integrate the dynamic requests into the existing route. Every time a dynamic request appears, the vehicle has to start from the depot to service the customer, which results in high routing costs. To prevent this interdependency, we partly follow the approach introduced in [15] and determined the time t_i for a dynamic request i as evenly distributed between 0 and the time horizon T. So, for all created points in time t_i for all dynamic requests n the following rule is valid: $t_{i+1} - t_i = d_i; d_i = d_{i+1} \forall i$, where $t_{n+1} = T$. T is determined by solving the static instance with the Jsprit framework [31]. To prevent the interdependency between the location of request i and the time t_i, we executed the simulation to evaluate the dynamic problem instance several times with different assignments between t_i and i. Additionally, we simulated the instances in a reversed routing order. For example, if a routing starts with servicing the static customer A, B, and ends at static customer C. Additional simulations in reversed order where the routing starts at customer C and ends with servicing customer A are performed. To generate one dynamic problem instance using SEREIN, we had to determine 6 sample routing costs with the help of simulation. The average of these routing costs is used by SEREIN to unfold the search space, identifying instances which are easy as well as instances which are hard to solve for the Greedy algorithm. SEREIN created 2,000 dynamic problem instances for each combination of $DOD \in \{0.1, 0.2, \ldots, 0.9\}$ and problem instance shown in Figure 3. This results in 72,000 dynamic problem instances which got evaluated with 432,000 simulations, only to create the problem space Y.

2.3 Algorithm and the Performance Space

For the Greedy algorithm, we already evaluated the Performance Space Y while creating the Problem Space P. The Greedy algorithm inserts a dynamic request, appearing during problem simulation, into the current planned route for the vehicle. We determined the initial route with the help of the Jsprit framework [31]. The Greedy algorithm evaluates all possible insertion points and chooses the one that produces the best solution costs. A detailed description of the algorithm can be found in [15]. Since we need at least two algorithms to evaluate our introduced DVRP features. We implemented a replanning approach, where every new dynamic request triggers a complete replanning of the tour. So both considered solution approaches are following very contrary ideas. The Re-planning algorithm, that we implemented uses the R package *tspmeta* from [14]. The implementation is based on a 2-opt optimisation algorithm, which was one of the first successful algorithms to solve larger TSP instances and which is still widely used in practice [14]. Due to the stochastic characteristics of the 2-opt approach, we had to simulate each dynamic problem instance (72,000) several times. Overall, we used additional 432,000 simulations (also considering simulations in reverse customer request order) to determine the Performance Space Y for the Re-planning algorithm.

2.4 Evaluation of the new DVRP Features

We separate the generated problem instances into a group of instances which are easy (low solution costs) and into a group of instances which are hard (high solution costs) to solve for each algorithm. We calculate the feature vectors for all instances and evaluate all possible feature combinations and their predictive power to differentiate between hard and easy instances. In this Section we present an exemplary excerpt from our evaluations.

Figure 4 shows the relation between feature *angle_mean* and *mst_dists_mean* from the first group of features (features considering dynamic customer demands only). The graphs in the first row show results derived from the problem instance CMT04. The suitability of the features to distinguish between easy and hard instances is visible by optically separable point clouds for all DOD's. With an increasing DOD, the point clouds are harder to separate from each other. The graphs in the second row consider all instances (compare Figure 3). Here, we observe more point clouds which are harder to separate visually. The borders between the clouds get more indistinct with a decreasing DOD. Similar patterns occurred for other feature combinations from the first group. The evaluation of the features from the second group (features considering static and dynamic customer demands) show similar characteristics, compare Figure 5. But in general there are better and less visually separable point clouds even for higher DOD's. The combination of features of both groups show its suitability to distinguished between hard and easy problem instance too, compare Figure 6.

In general the feature evaluation shows, that the introduced features are suitable to distinguish between easy and hard problem instances for both applied algorithms. Note, that an important feature to distinguish between instances is the DOD, only implicitly shown in the graphs.

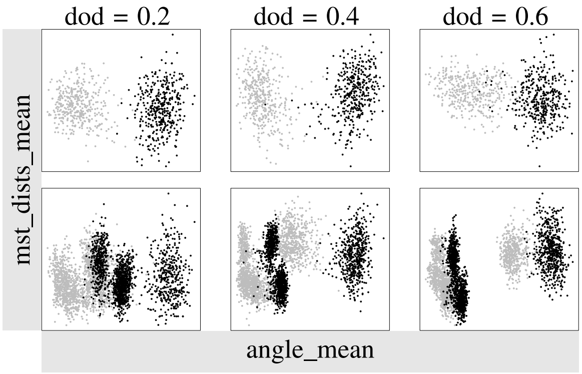

Figure 4: Evaluation of feature combination *angle_mean* and *mst_dists_mean* for Re-planning algorithm.

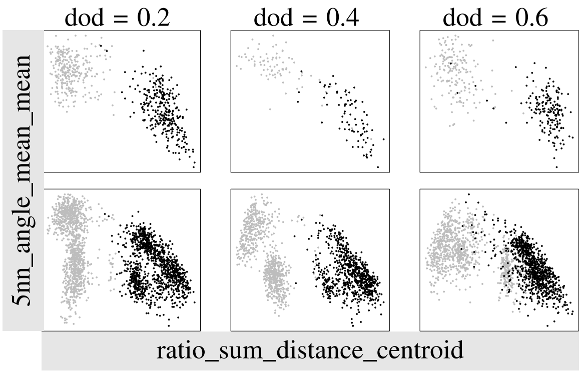

Figure 5: Evaluation of feature combination *ratio_sum_distance_centroid* and *5nn_angle_mean_mean* for Re-planning algorithm.

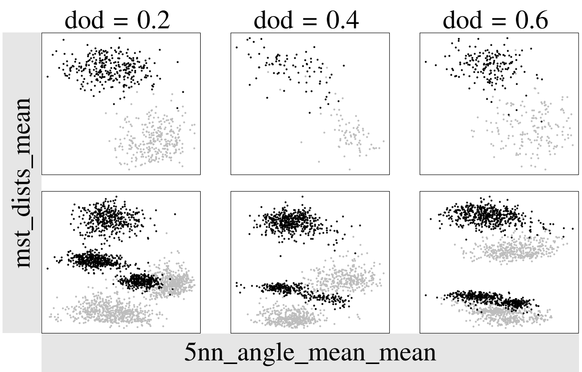

Figure 6: Evaluation of feature combination *mst_dists_mean* and *5nn_angle_mean_mean* for Re-planning algorithm.

3 Algorithm Selection

To test our problem features in the context of Algorithm Selection, we implemented a regression machine learning technique to model the relationship between the problem features and the algorithm performance. We used Artificial Neural Network (ANN) which map unlabeled input to a label (output) using internal data structures. We standardise all values of the feature vectors using the standard scalar method implemented in [32]. These standardised feature vectors are the input for the model (balanced regrading instances and DOD). The output for the regression model is the standardised routing costs for the instances. For each algorithm (Greedy, Re-planning) we constructed one regression model. Due to its popularity we used Tensorflow from Google Brain introduced in [33] for the implementation. Both ANNs have 4 hidden layers with 60, 18, 20, and 22 neurons. The input layer has 184 neurons and the output layer has 1. All neurons are using the activation function RELU, [34]. The structure of the model was setup using systematic trial and error, a common approach to approximate the optimal number of hidden layers and nodes, [35]. All neurons of each layer are fully connected to all neurons of the following layer. We applied the optimisation function *Adam* introduced in [36] to determine the weights of the edges between the neurons of the ANNs. Beside testing our approach with unseen problem instance, we applied the grouped cross validation method to validate our trained ANNs. The method divides the available data into k groups and constructs k different ANNs using $k-1$ data groups. The k^{th} group is used for testing the constructed ANN, [37]. The error is determined using the mean of testing set errors of the groups. The grouped cross validation method is currently considered as the gold standard for testing ANNs. Note, that we had to ensure that the combination of instance and DOD are evenly distributed within all created groups. For our validation we defined 5 data groups. The results of the validation of our constructed ANNs are shown in Table 3. The applied metric for the regression model is the mean squared error (MSE), see for example, [38]. The results in Table 3 show that our features are very suitable for performance prediction. The average error for both ANNs is less than half a percent for all cross validation groups. The models are not over-fitted and precisely predict the algorithm performance, even for unseen dynamic instances generated from the static instances shown in Figure 3.

Algorithm	Mean square error (+/- Standard deviation)
Greedy	0.006775 (+/- 0.000126)
Re-planning	0.003549 (+/- 0.000105)

Table 1: Grouped cross validation results (5 groups) of our constructed ANNs.

To test our regression models with completely unseen problem instances, we created dynamic instances from a subset of VRP instances introduced in [39] and [40], see Figure 7. We created 500 random instances for each instance for each $DOD \in \{0.3, 0.4, \ldots, 0.8\}$ and determined the routing costs with the Greedy and the Re-planning algorithm. The sum of the routing costs for each algorithm, $R = \sum_{i=0}^{n} r_{algorithm.i}$, is shown in Table 3 (column Greedy and Re-planning). Based on the known routing costs for the Greedy and for the Re-planning algorithm, we have been able to determine the correct algorithm selection for each problem instance. The correct selected algorithm is the one which produces the lower costs. The sum of the routing costs, based on a correct algorithm selection, $Correct = \sum_{i=0}^{n} min(r_{greedy.i}, r_{replan.i})$, is also shown in Table 3 (column Correct). Additionally Table 3 shows the sum of the routing costs based on our algorithm selection with our regression (column Regression) models. In general the results, shown in Table 3, illustrate that we are able to successfully select algorithms for unseen problem instances with our approach. Our algorithm selection approach performs better, compared to exclusive use of either the Greedy or the Re-planning algorithm in most cases. However, for the dynamic problem instances derived from problem instance A-n60-k9, our algorithm selection leads to worse results in comparison to the pure Re-Planning algorithm.

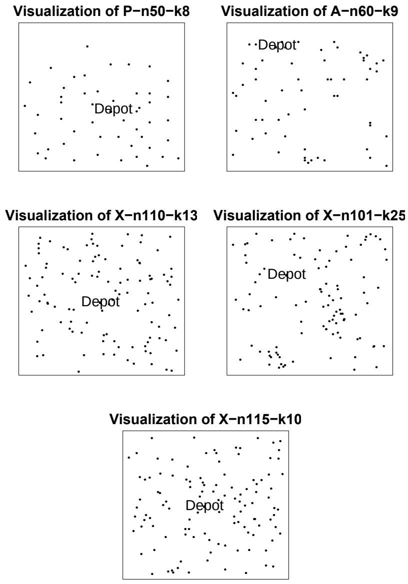

Figure 7: Subset of VRP instances introduced in [39] (P-n50-k8, and A-n60-k9) and [40] (X-n110-k13, X-n101-k25, and X-n115-k10). The dots represent the customer requests.

4 Conclusion and Outlook

Our results show that our approach is able to reliably autonomously select the better performing algorithm for various known and partly also for unseen DVRP instances. Based on simulation results we have been able to derive characteristic DVRP from static problem instances. We developed and calculated features for these instances. Our introduced features in combination with the simulation results are the bases for our developed regression models which we trained to predict algorithm performance. The prediction of algorithm performance is the base for our successful autonomous algorithm selection.

For our performance prediction models we are currently considering only 4 different structured problem instances, compare Figure 3. The trained regression models are quite fitted to these instances which seems to be the reason for the difficulties with performance prediction for the unseen problem instances A-n60-k9, compare Table 3. The idea is to build a stronger basis for our performance prediction by considering more different kinds of problem instances. We plan to include the very geometrical structured VRP instances introduced in [41] in our research. We also have to evaluate the possibility to create own artificial instances.

Currently we only consider two different algorithms which perform similar for different problem instances, compare Section 3. Future work will be the training and testing of regression models based on simulation results of different DVRP algorithms. We will implement an DVRP algorithm portfolio with existing algorithms. We are currently developing a new DVRP solving approach based on the results from [15]. Beside considering more instances and more algorithms, the question occurs which evolved instances are good for training

3000 Instances based on	Greedy	Re-planning	Correct	Regression
X-n110-k13	31,664,462	31,836,229	*31,405,740*	**31,413,135**
X-n101-k25	34,794,388	34,725,471	*34,389,425*	**34,455,935**
X-n115-k10	39,754,906	39870583	*39402544*	**39,429,201**
P-n50-k8	534,005	534,690	*528,157*	**533,961**
A-n60-k9	2,787,325	**2,781,479**	*2,752,555*	2,783,146

Table 2: Validation results of our constructed ANNs for unseen instances.

and testing purposes. Currently we are not verifying if an evolved instances is a good representation for the performance class. In future work, we have to determine a metric which provides information about how good a certain instance represents a certain algorithm performance.

References

[1] Dantzig GB, Ramser JH. The Truck Dispatching Problem. *Management science*. 1959;6(1):80–91.

[2] Flood MM. The traveling-salesman problem. *Operations Research*. 1956;4(1):61–75.

[3] Lahyani R, Khemakhem M, Semet F. Rich Vehicle Routing Problems: From a Taxonomy to a Definition. *European Journal of Operational Research*. 2015; 241(1):1–14.

[4] Pillac V, Gendreau M, Guéret C, Medaglia AL. A Review of Dynamic Vehicle Routing Problems. *European Journal of Operational Research*. 2013; 225(1):1–11.

[5] Psaraftis HN, Wen M, Kontovas CA. Dynamic vehicle routing problems: Three decades and counting. *Networks*. 2016;67(1):3–31.

[6] Wilson NH, Colvin NJ. *Computer Control of the Rochester Dial-a-ride System*. 77-22. Cambridge, Massachusetts. 1977.

[7] Ritzinger U, Puchinger J, Hartl RF. A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*. 2016; 54(1):215–231.

[8] Clarke G, Wright JW. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations research*. 1964;12(4):568–581.

[9] Juan AA, Faulin J, Jorba J, Riera D, Masip D, Barrios B. On the use of Monte Carlo Simulation, Cache and Splitting Techniques to Improve the Clarke and Wright Savings Heuristics. *Journal of the Operational Research Society*. 2011;62(6):1085–1097.

[10] Juan AA, Faulin J, Grasman SE, Rabe M, Figueira G. A Review of Simheuristics: Extending Metaheuristics to Deal with Stochastic Combinatorial Optimization Problems. *Operations Research Perspectives*. 2015; 2:62–72.

[11] Gruler A, Klüter A, Rabe M, Juan AA. A Simulation-Optimization Approach for the Two-Echelon Location Routing Problem Arising in the Creation of Urban Consolidation Centres. In: *Simulation in Produktion und Logistik 2017*, edited by Wenzel S, Peter T. Kassel: kassel university press. 2017; pp. 129–138.

[12] Wolpert DH, Macready WG. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*. 1997;1(1):67–82.

[13] Rice JR. The Algorithm Selection Problem. *Advances in Computers*. 1976;15:65–118.

[14] Mersmann O, Bischl B, Trautmann H, Wagner M, Bossek J, Neumann F. A Novel Feature-based Approach to Characterize Algorithm Performance for the Traveling Salesperson Problem. *Annals of Mathematics and Artificial Intelligence*. 2013;69(2):151–182.

[15] Mayer T, Uhlig T, Rose O. A Location Model for Dynamic Vehicle Routing Problems. In: *Simulation in Produktion und Logistik 2017*, edited by Wenzel S, Peter T. Kassel: kassel university press. 2017; pp. 149–158.

[16] Uhlig T. *Self-Replicating Individuals*. München: Verlag Dr. Hut. 2015.

[17] Mayer T, Uhlig T, Rose O. An open-source discrete event simulator for rich vehicle routing problems. In: *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, edited by Rossetti R, Wolf D. New York, USA: IEEE. 2016; pp. 1305–1310.

[18] Smith-Miles K, Lopes L. Measuring Instance Difficulty for Combinatorial Optimization Problems. *Computers & Operations Research*. 2012;39(5):875–889.

[19] Smith-Miles KA. Cross-disciplinary Perspectives on Meta-learning for Algorithm Selection. *ACM Computing Surveys (CSUR)*. 2009;41(1):6.

[20] Smith-Miles K, van Hemert J, Lim XY. Understanding TSP Difficulty by Learning from Evolved Instances. In: *Proceedings of the 4th International Conference on Learning and Intelligent Optimization*, edited by Blum C, Battiti R. Berlin, Heidelberg: Springer-Verlag. 2010; pp. 266–280.

[21] Lin S, Kernighan BW. An Effective Heuristic Algorithm for the Traveling-salesman Problem. *Operations Research*. 1973;21(2):498–516.

[22] Mersmann O, Bischl B, Bossek J, Trautmann H, Wagner M, Neumann F. Local Search and the Traveling Salesman Problem: A Feature-Based Characterization of Problem Hardness. In: *Learning and Intelligent Optimization*, edited by Hamadi Y, Schoenauer M, pp. 115–129. Berlin, Heidelberg: Springer Berlin Heidelberg. 2012;.

[23] Wagner M, Lindauer M, Misir M, Nallaperuma S, Hutter F. A Case Study of Algorithm Selection for the Traveling Thief Problem. *Journal of Heuristics*. 2018; 24(3):295–320.

[24] Cheeseman P, Kanefsky B, Taylor WM. Where the Really Hard Problems Are. In: *Proceedings of the 12th International Joint Conference on Artificial Intelligence - Volume 1*, edited by Mylopoulos J, Reiter R. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. 1991; pp. 331–337.

[25] Ridge E, Kudenko D. An Analysis of Problem Difficulty for a Class of Optimisation Heuristics. In: *Proceedings of the 7th European Conference on Evolutionary Computation in Combinatorial Optimization*, edited by Cotta C, Van Hemert J. Berlin, Heidelberg: Springer-Verlag. 2007; pp. 198–209.

[26] Kanda J, Carvalho A, Hruschka E, Soares C. Selection of algorithms to solve traveling salesman problems using meta-learning 1. *International Journal of Hybrid Intelligent Systems*. 2011;8(3):117–128.

[27] Mersmann O, Bischl B, Trautmann H, Preuss M, Weihs C, Rudolph G. Exploratory landscape analysis. In: *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM. 2011; pp. 829–836.

[28] Lund K, Madsen OB, Rygaard JM. Vehicle Routing with Varying Degree of Dynamism. 1996; (IMM-REP-1996-1).

[29] Larsen A. The Dynamic Vehicle Routing Problem. Ph.D. thesis, Institute of Mathematical Modelling, Technical University of Denmark. 2000.

[30] Christofides N. The Vehicle Routing Problem. *Revue française d'automatique, informatique, recherche opérationnelle Recherche opérationnelle*. 1976; 10(V1):55–70.

[31] Schröder. Jsprit. `https://github.com/graphhopper/jsprit`. 2018. Accessed July 2nd, 2018.

[32] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al. Scikit-learn: Machine learning in Python. *Journal of machine learning research*. 2011; 12(11):2825–2830.

[33] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, Kudlur M, Levenberg J, Monga R, Moore S, Murray DG, Steiner B, Tucker P, Vasudevan V, Warden P, Wicke M, Yu Y, Zheng X. TensorFlow: A System for Large-scale Machine Learning. In: *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, OSDI'16. Berkeley, CA, USA: USENIX Association. 2016; pp. 265–283.

[34] Nair V, Hinton GE. Rectified Linear Units Improve Restricted Boltzmann Machines. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML)*, edited by Fürnkranz J, Joachims T. USA: Omnipress. 2010; pp. 807–814.

[35] Basheer IA, Hajmeer M. Artificial Neural Nnetworks: Fundamentals, Computing, Design, and Application. *Journal of microbiological methods*. 2000;43(1):3–31.

[36] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. 2015;.

[37] Twomey JM, Smith AE. *Validation and Verification*. New York, NY, USA: American Society of Civil Engineers. 1997.

[38] Lehmann EL, Casella G. *Theory of Point Estimation*. 175 Fifth Avenue, New York: Springer Science & Business Media, 2nd ed. 2006.

[39] Christiansen CH, Lysgaard J. A Branch-and-price Algorithm for the Capacitated Vehicle Routing Problem with Stochastic Demands. *Operations Research Letters*. 2007;35(6):773–781.

[40] Uchoa E, Pecin D, Pessoa A, Poggi M, Vidal T, Subramanian A. New Benchmark Instances for the Capacitated Vehicle Routing Problem. *European Journal of Operational Research*. 2017; 257(3):845–858.

[41] Golden BL, Wasil EA, Kelly JP, Chao IM. The Impact of Metaheuristics on Solving the Vehicle Routing Problem: Algorithms, Problem Sets, and Computational Results. In: *Fleet Management and Logistics*, edited by Crainic TG, Laporte G, pp. 33–56. Boston, MA: Springer US. 1998;.

Expansion of Models for Heart Rate Variability beyond the Autonomic Nervous System

Jennifer Straub[1*], Martin Bachler[2]

[1]Institute of Analysis and Scientific Computing, TU Wien,
Wiedner Hauptstraße 8-10,1040 Vienna, Austria; *jennifer.straub@tuwien.ac.at

[2] AIT Austrian Institute of Technology GmbH, Center for Health & Bioresources, Biomedical Systems,
Giefinggasse 4, 1210 Vienna, Austria

SNE 29(4), 2019, 189-198, DOI: 10.11128/sne.29.tn.10494
Received:Sept. 15, 2019; Revised: November 5, 2019;
Accepted: November 10, 2019
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna,
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. According to the World Health Organisation, diseases of the cardiovascular system (CVS) are currently the main cause of death all over the world. Therefore, their understanding, prediction, and prevention with the help of non-invasive, cost effective methods is of great interest. Analysis of the heart rate and its change over time can give valuable insight into the health status of a patient, and is easily derived from electrocardiogram (ECG) data. Reduced heart rate variability (HRV) is associated to an increased probability of dying after myocardial infarctions and indicates inflammatory processes. It is symptomatic of mental disorders such as depression and burn-out. Different approaches in modeling and simulation of HRV can provide new insight into the non-linear interplay of cardiovascular regulation. In this work, three models for HRV are implemented and compared. They include the firing rate of the baroreceptors, respiration, activity of the sympathetic and parasympathetic nervous system, stroke volume, cardiac noradrenaline and acetylcholine concentration, as well as a windkessel model including peripheral resistance and arterial compliance. First, an existing model for HRV based on respiration and baroreflex activity was implemented and analyzed. A second model was created through adaption of the first model. Based on a model for the autonomic response to orthostatic stress, a third model was implemented as well. All models were realized in Simulink 2017b, and their validation is performed based on 60 five-minute ECG recordings from 30 subjects. The simulation results are compared to subject data based on the standards of HRV measurement by the Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology. Each of the three modeling approaches showed specific advantages, disadvantages, and possibilities for further improvement. The results provide basis for extension of HRV models, paving the way for the future usage of model prediction in the field of cardiovascular diseases.

Introduction

Heart rate variability (HRV), the change in time intervals between successive heart beats, reflects the activity of the autonomous nervous system, and gives information about the overall condition of the cardiovascular system. The heart rate (HR) keeps on changing as a result of non-static physiological and psychological regulatory mechanisms, which interact in a non-linear way. HRV is a sign of a healthy cardiovascular system. On the contrary, a reduced change between adjacent interbeat times can indicate health problems such as coronary artherosclerosis, inflammation, and depression. HRV decreases with age, and depends on the HR, since a higher HR results in shorter interbeat intervals, and therefore leaves less time for changes [1].

As HRV can be easily derived from electrocardiogram (ECG) data and therefore is a simple and non-invasive diagnostic tool, great interest lies in attaining a deeper understanding of its nonlinear regulatory mechanisms, using modeling and simulation. Creating a model which is complex enough to reproduce important characteristics of HRV, but still simple enough to maintain applicability, can lead to a deeper physiological insight [2], opening up new possibilities for diagnostics and therapy [3].

1 Background

1.1 Controlling Mechanisms of the Cardiovascular System

The Autonomic Nervous System The autonomic nervous system (ANS) has two antagonistic branches: the sympathetic and the parasympathetic nervous system. The sympathetic system is mainly activated during exercise and emergency situations to increase HR and blood flow. The parasympathetic system is dominant during rest conditions. It regulates basic body funcions and decreases the HR [4]. The excitement of their effector organs happens via the release of acetylcholine or noradrenaline [5].

Efferent nervous activity of the ANS is mostly regulated by autonomic reflexes, for example the baroreceptor reflex[4]. The tissues of the heart are innervated by both systems, but parasympathetic fibres are mainly distributed to the atria, the sinus node, the AV-node, and not to the ventricles. Through sympathetic stimulation, noradrenaline is released regardless of the stretch of the muscle fibres, causing stronger contractions, as well as faster depolarisation and relaxation, resulting in a higher HR. Parasympathetic activity antagonistically prevents the release of noradrenaline from sympathetic nerves. The parasympathetic HR reduction also comes from a release of acetylcholine at the heart's pacemaker cells. The release of sympathetic noradrenaline also causes the blood vessels in the body to contract, which regulates blood pressure [6].

Continuous discharge of sympathetic nerve fibres increases pumping by 30 percent compared to no sympathetic stimulation. The average HR for young adults of 70 beats per minute (bpm) can be increased up to approximately 200 bpm by strong sympathetic stimulation, while simultaneously intensifying the contraction to as much as double normal, increasing the amount of blood pumped and augmenting the ejection pressure [6].

The Baroreflex The baroreceptors are nerve endings located at the sinus caroticus and in the aortic arch. They almost immediately generate nervous signals when stretched by high blood pressure. The signals are then forwarded to the brain stem. From there, secondary signals increase the activity of the parasympathetic center. This leads to a decrease in HR, a weaker heart contraction, and vasodilation.

Conversely, low blood pressure has the opposite effect, causing oscillation of the baroreceptor reflex. If arterial blood pressure is chronically increased or decreased, baroceptors activity changes at first, but adapts to the new pressure level in 1 to 2 days[4].

The Arterial Windkessel Effect The wall of the thoracic aorta contains a lot of elastic fibres. Therefore, the blood ejected from the heart during systole causes the aorta to stretch, which reduces the systolic pressure rise and subsequently has an effect on the baroreceptors. The aorta then returns to its original shape, pushes the blood further into the arterial system and ensures a rather continuous blood flow [5].

The Effect of Respiration on the Heart Respiratory sinus arrhythmia (RSA) is the synchronous variation of HR and respiration. The stroke volume of the heart is almost equivalent to the blood volume in the capillary bed, and as a consequence, variation of the HR during inspiration and expiration affects the efficiency of the gas exchange[7].
During inspiration, the intervals between successive heartbeats are shortened, whereas during expiration longer intervals are observed. RSA can be influenced by cardiopulmonary function, sleep or wakefulness, age, and many other factors [7].

1.2 State of the Art

Existing models include various components of the cardiovascular system and make use of highly statistical approaches [8, 9], as well as differential equations [10], and discrete events [11], but a lack of detailed physiological reasoning is found in a lot of them.

There is broad consensus about the physiological reasonableness of the integral pulse frequency modulation (IPFM) model with constant or varying threshold for the generation of a heart beat time series [12, 13].

A simplistic approach for including the effect of respiration on the HR is presented by Brennan et al. [14]. Although this approach is physiologically reasonable, respiratory influences on the HR are usually included into the parasympathetic activity[15].

Aside from respiration, the baroreflex is often part of HRV models. Olufsen et al. present a model of baroreflex regulation of the HR during orthostatic stress, based on different types of baroreceptors [16].

Instead of including more effectors than the baroreflex, Ursino proposed a compartment model, which includes a more detailed description of the left and right heart, the pulmonary circulation, and their respective resistances and compliances, which are modulated by the activity of the ANS [17].

DeBoer, Karemaker, and Strackee [11] created a more extensive model based on difference equations, amongst other factors including an RR-interval dependent contractility of the myocard.

A closed loop model, which includes respiration, the baroreceptor reflex, ANS activity, a windkessel time constant, contractility, and even neurotransmitters levels, is proposed by Seidel and Herzel. Opposite to the previously mentioned model, it combines discrete signals and continuous ones. A unique characteristic of this model is the use of a phase response curve, which modulates the input of the parasympathetic nervous system depending on its time of occurrence during the heart cycle [2].

2 Methods

Integral Pulse Frequency Modulation Model Many authors assume a modulating effect of the ANS activity on the sino-atrial node[12]. This influence is summarized as a function $m(t)$ with zero mean and a rather small amplitude. The beat occurrence times t_k generated by the Integral Pulse Frequency Modulation (IPFM) Model can then be written as

$$k = \int_0^{t_k} \frac{1+m(t)}{T} dt \qquad k = 1,2,3..., \tag{1}$$

Poincaré Plots Poincaré plots allow the geometrical analysis, and quantification of HRV by plotting each RR-interval against the following one, resulting in a scatterplot. It simultaneously gives an overview of the overall, short and long term beat-to-beat variability[18]. To characterize the shape of a Poincaré plot, the ellipse fitting technique is used. Other Statistical measures for short and long term variability are SD1 and SD2 as described by Brennan [19].

3 Models

3.1 Model 1: An HRV model including respiration and baroreflex

As a first nonlinear model for HRV, presented by Henrik Seidel and Hanspeter Herzel including baroreceptors, ANS activity, neurotransmitters, contractility, vasoconstriction, and a windkessel, depicted in Figure 1, was implemented. All parameters were chosen according to the original publication [2].

3.2 Model 2: An Adapted Seidel and Herzel Model

The following paragraphs describe a second model, based on the previously presented one, but with the aim of creating a more physiologically accurate model.

Blood pressure and systolic duration The blood pressure curve generated by the previous model shows a sharp peak at the end of the systole, followed by a steep exponential decay. If ever, a discontinuity in the time derivative of the blood pressure function, as seen at the end of the systole, should occur when the aortic valves close due to a lack of output from the left ventricle. The systolic duration τ_{sys} was estimated at 125ms, which is clearly too short compared to results from in vivo studies, and was therefore set to 300ms [20]. The original equation for systolic pressure was replaced by the following:

$$p_I = d_{i-1} + S_i \left(\frac{1.8(t-t_i)}{\tau_{sys}} \right) \exp\left(1 - \frac{1.8(t-t_i)}{\tau_{sys}} \right). \tag{2}$$

Sympathetic activity The original equation for sympathetic activity was modified to avoid discontinuities in the first derivative. This leads to the following expression:

$$\nu_s = \max\left(0\,,\ \nu_{s,0} - k_{s,b}\nu_b + k_{s,r}\frac{\sin(\pi s_f t + \Delta\phi_{s,r}) + 1}{2} \right). \tag{3}$$

A similar problem occurs due to the subtraction of $k_{s,b}\nu_b$. Therefore, a butterworth low-pass filter is introduced, using order 3 and a passband edge frequency of $0.15 \cdot 2\pi \frac{rad}{sec}$. The filter output is multiplied by 0.4, to keep a reasonable level of sympathetic activity.

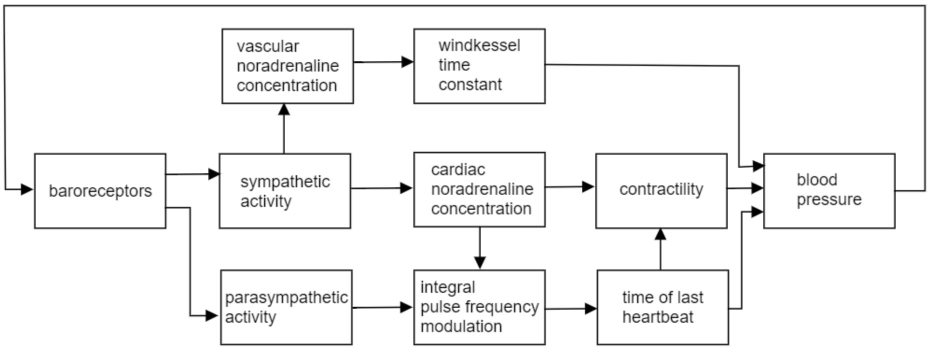

Figure 1: Schematic representation of the CVS components in Model 1 and 2.

Parasympathetic activity and respiration Respiration is only included in the parasympathetic activity [21]. Assuming that the breathing pattern is represented by a function $r(t)$, ranging from -1 at total expiration to 1 at total inspiration, a linear dependence of parasympathetic activity on respiration is assumed. To keep the same overall level of activation, the original equation was changed to

$$\nu_s = \max\Bigg(0\ ,\ \nu_{p,0} + k_{p,b}\nu_b + k_{p,r}\left(\frac{\sin(\pi p_f t + \Delta\phi_{p,r})}{2} + \frac{r(t)}{2}\right)\Bigg), \tag{4}$$

including a basic parasympathetic frequency p_f.

Correction of the mean HR To accurately reproduce the mean HR defined by T_0, the product of the sympathetic and parasympathetic activities f_s and f_p should equal 1 in the long term as part of the IPFM input. To ensure this, the mean of $f_s(t)f_p(t)$ at time t was divided by its mean over the previous 10 sec.

3.3 Model 3: An HRV model with different baroreceptor types

In this section, the model presented by Olufsen et al. is described and extended. It was designed to derive ANS activity and thus HR from measured blood pressure values during periods of orthostatic stress. Most parameters, such as τ_S, τ_I, τ_L, k_S, k_I, k_L, and M were derived from animal experiments, others were estimated empirically [16].

Baroceptors Olufsen et al. distinguish three types of baroreceptors. The model is based on the rate of change of mean arterial pressure $\overline{p}$ over time. Assuming that the pressure function $p(s)$ is continuous and bounded, by using the integral rule of Leibniz there holds:

$$\frac{d\overline{p}}{dt} = \alpha\left(\frac{d}{dt}\int_{-\infty}^{t} p(s)e^{-\alpha(t-s)}ds\right) = \alpha(p-\overline{p}). \tag{5}$$

Three types of baroreceptors are defined as

$$\frac{dn_i}{dt} = k_i\frac{d\overline{p}}{dt}\frac{n(M-n)}{(M/2)^2} - \frac{n_i}{\tau_i} \qquad i = S,I,L \tag{6}$$

S stands for short, I for intermediate, and L for long time scales. They represent the variations in threshold for different baroreceptor types and are included in the equations via τ_S, τ_I, and τ_L. The maximum firing rate is taken into account by M, which is set to 120. For the overall firing rate n, there holds $n = n_S + n_I + n_L + N$ and n_S, n_I and n_L give the deviation from the mean firing rate N.

The autonomous nervous system The parasympathetic activity $T_p(n)$ is defined as n divided by M. The sympathetic activity includes a time-delay τ_d and is given as:

$$T_s(n) = \frac{1-n(t-\tau_d)/M}{1+\beta T_p(n)}. \tag{7}$$

The Heart as effector of the ANS Cardiac noradrenaline and acetylcholine concentrations are calculated as

$$\frac{dC_{nor}}{dt} = \frac{-C_{nor} + T_s}{\tau_{nor}}, \quad \frac{dC_{ach}}{dt} = \frac{-C_{ach} + T_p}{\tau_{ach}}, \qquad (8)$$

and to generate an input for the IPFM, they were combined with the mean HR H_0 giving

$$\frac{d\varphi}{dt} = H_0(1 + M_s C_{nor} - M_p C_{ach}). \qquad (9)$$

In order to create a closed loop, the previously presented model had to be further expanded.

The Arterial Windkessel A three-element windkessel model was chosen to generate a pressure curve as input for the baroreceptors. It is dependent on total arterial compliance C, peripheral resistance R and input impedance R_c and was chosen accoring to Westerhof et al. [22].

In order to solve the given differential equation, a function for the flow from the aortic root $q(t)$ was chosen based on findings in [23] as

$$q(t) = exp\left(2\left(1 - \frac{1}{t_s}\right) + \frac{1}{(\frac{2t}{t_s-1})^2 - 1}\right)\frac{SV}{I} \qquad (10)$$

The systole duration in ms is calculated as $t_s = 540 - 2.1 \cdot HR$ according to [24]. SV is the stroke volume in milliliters eqaling $n = 0.5 + 0.5\frac{RR_{n-1}V_{ref}}{1000}$ according to [25]. At 60bpm it was assumed to be 70ml. I equals the integral of the exponential term in $q(t)$ over the interval $[0; t_s]$ [23].

Autonomic nervous system activity Two sinusoidal functions with frequencies f_p and f_s were added to the model, giving

$$T_p(n) = \frac{n(t)}{M} + C_p \sin(2\pi f_p), \qquad (11)$$

$$T_s(n) = \frac{1 - \frac{n(t-\tau_d)}{M} + u(t)}{1 + \beta\frac{n(t)}{M}} + C_s \sin(2\pi f_s). \qquad (12)$$

To avoid fast fluctuations in sympathetic nervous system activity, a butterworth filter of order 3 and passband edge frequency of $0.15 \cdot 2\pi$ was applied to $T_s(n)$.

IPFM and Respiration The ANS input of the IPFM model was scaled analogously to the second model. The respiration was included by using a varying threshold for the IPFM, as proposed by Barbi et al. [26]. The threshold is chosen as $i(t) = 1 + \frac{r(t)}{20}$ with $r(t)$ oscillating between -1 and 1, depending on the breathing cycle.

4 Simulation

All three simulations were implemented in Simulink 2017b and the RR-intervals were further analysed in Matlab 2017b.

RR-interval data from 30 subjects with essential hypertension was used to derive parameters for the simulation. One of six different guided breathing patterns was used for each subject. During the 10min breathing exercise, an ECG was recorded at a sampling rate of 256 Hz [27].

The 10min were split into two 5min recordings and hence 60 different data sets were used for the simulation runs of each mode. For model 2 and 3, basic frequencies for the sympathetic and the parasympathetic branch of the ANS needed to be determined. Therefore, a Lomb-Scargle periodogram of the patients RR-intervals was calculated. The sympathetic frequency is chosen as the frequency with the highest power in the interval from $0.04 - 0.15$ Hz. The same applies for the parasympathetic activity and the frequency band from $0.15 - 0.4$ Hz.

For the solution of the various non-stiff differential equations, the ode45 solver with variable step size was used. Every simulation was performed for 1000 seconds, from which only the last 300 seconds were selected.

5 Results

Poincaré Plots The Poincaré plots of the first models 60 simulation runs clearly show a tendency of overestimation of the mean HR. This is apparent through the shift of the whole set of points along the line of identity, compared to the given patient data.

Another significant mismatch between given data and simulated data lies in the general shape of the point cloud. A typical distribution of points has a comet-like shape, whereas the simulated data presents itself in a more circular shape without any points in the centre of the point cloud.

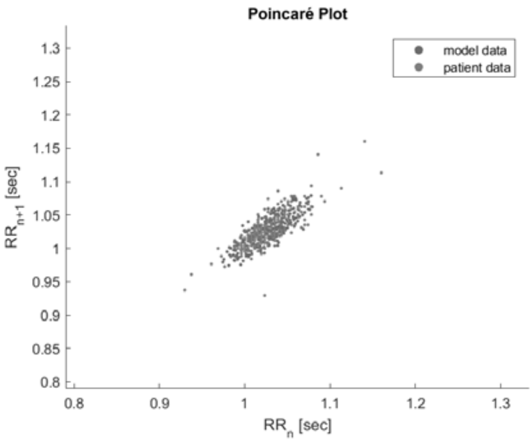

Figure 2: Model 2: Poincaré plot of the first 5min recording of subject number 23, with corresponding simulation output.

For the second model, considerably better Poincaré plots could be achieved, especially in relation to the mean HR (Figure 2). For some cases however, the simulation output resulted in an atypical, elliptically appearing line. This phenomenon occurred mostly in connection with input data showing a rather large general dispersion of points.

The third model showed similar results concerning the mean HR, but a tendency of overestimation of the plots length and width was observed. For a variety of data sets, the plot showed a lack of points in the centre of the point cloud, although far less pronounced in comparison to the first model.

Statistical HRV Analysis Six statistical parameters were used for the quantification of the RR-data [28, 19]. The mean RR- duration was analysed, followed by SD1, RMSSD, and SDSD as indicators for short term variability, SD2 as quantifier for long term variability and SDNN, which represents overall HRV.

As already indicated, model 1 tends to overestimate the RR-interval length drastically. For model 2 and 3, a very good replication of mean RR-duration could be reached. As shown in Figure 3, the results for the SDSD show a similar pattern to that of the mean RR-intervals. Again, the first model clearly overestimates the SDSD, opposite to the second and third model, which show a considerably better fit. When looking at the differences it should be noted, that the second model shows a slight underestimation, whereas the third model overestimates the SDSD a little.

SD1 shows the same behaviour as SDSD concerning the boxplot. Model 1 and 2 show a comparable amount of underestimation of SD2, which quantifies slow HRV changes, as opposed to a considerable overestimation by model 3. Again this behaviour is also observable in the Poincaré plots of model 3, as an overly long extension of the model data along the line of identity.

The RMSSD, which reflects high frequency components, basically shows the same behaviour as the previously described measures of short term variability.

SDNN, which is a measure of total spectral power, is most accurately represented by the output of the first model, slightly underestimated by the second model and clearly overestimated by the third.

Blood Pressure For Model 1 and 2, blood pressure showed a highly unusual behaviour for a remarkable number of cases. Systolic blood pressure mostly ranged above 140 mmHg, reaching maximum values of up to 190mmHg.
This behaviour has a severe impact on the sympathetic activity, which is mainly dependent on the blood pressure function and its first derivative. Therefore, once systolic blood pressure levels exceed 160mmHg, sympathetic activity shows phases of zero activity or even vanishes completely.
Another noteworthy result are the rapid changes in systolic, as well as diastolic blood pressure over the course of a few successive heartbeats. The systolic blood pressure varies by up to 20mmHg for both models, although less pronounced for the first than the second model. This is due to the impact of sympathetic nervous activity on the cardiac noradrenaline concentration.

Another cause of blood pressure fluctuations of about 10mmHg in both models is presented by respiratory activity. The changes in pressure are time-delayed in a way, that the peak in blood pressure appears shortly before a maximum level of inhalation is reached.

For the third model, during none of the simulation runs, a systolic blood pressure higher than 130mmHg was reached, and it generally showed only small fluctuations (1-5mmHg) over time. For the diastolic blood pressure, normal values were observed, with two exceptional cases of 40mmHg and 50mmHg. In contrast to the systolic blood pressure, diastolic values showed quick changes of up to 10mmHg over the course of only a few heartbeats.

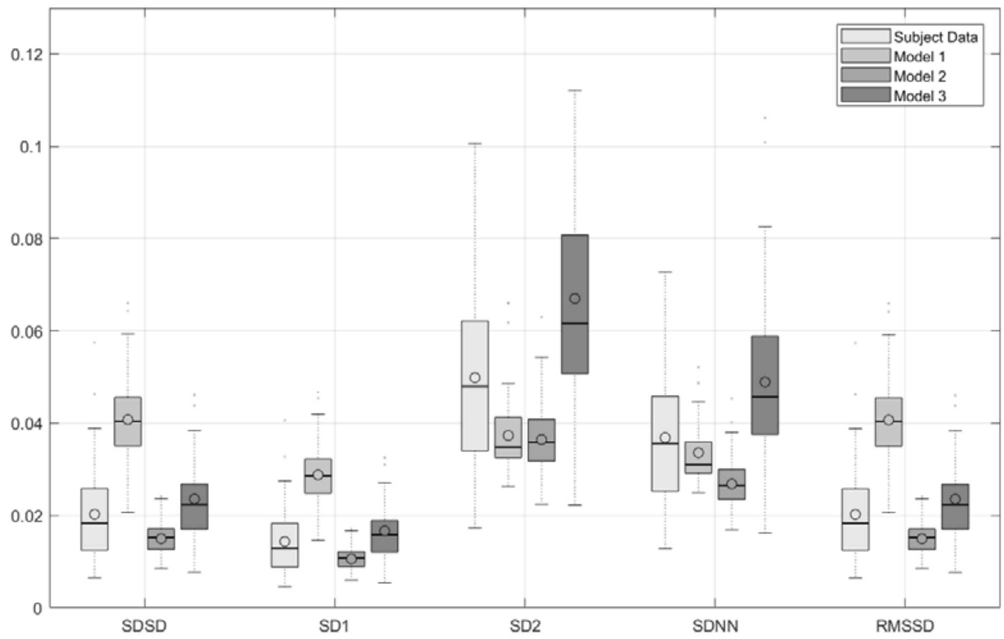

Figure 3: Boxplot of SDSD, SD1, SD2, SDNN, and RMSSD of the subject data and all three models.

Another notable observation about the behaviour of the blood pressure is its dependence on respiration. For the third model, respiration is included via variation of the IPFM threshold. It has an instantaneous influence on the occurrence time of the next heart beat, and consequently on blood pressure. For the first and second model, blood pressure variations can be observed, but due to the fact that respiration is included into the models ANS part, the time delay in sympathetic and parasympathetic activity results in a time delay between respiration and blood pressure variations.

6 Discussion

The Integral Pulse Frequency Modulation Model The first and most obvious result was the significant overestimation of the mean HR by the first model. This may be due to the fact that, although IPFM models are widely used in literature, they neglect respiration as an input, and their input functions are often designed to mimic physiological processes, but are not further mathematically analysed as stated by Meste et al. [29]. The averaged ANS IPFM input should equal 1 over the whole simulation run, in order to correctly recreate the mean HR. Mean HR should be additionally considered when modeling HRV, as it correlates with HRV [30].

The third model showed a similar problem. For the second and third model, the introduction of a scaling function did indeed improve the results considerably, but this approach does not address the root of the problem.

Still, if the mean amplitude of the IPFM model input is, for example, too low, and therefore upscaled, the amplitude of the periodic input components is also increased, resulting in greater fluctuations. The change in amplitude of periodic signal parts has notable effects on the shape of the Poincaré plot and the measures of HRV. For the third model, the input of the IPFM was generally underestimated. By upscaling it, fluctuations of the incoming signal were reinforced, resulting in a greater overall variability. Moreover, the low frequency input signal components are dampened by this kind of signal modification, which makes the approach unsuitable for use in modeling and simulation of HRV over longer periods of time.

The Baroreceptors Seidel and Herzel state, that it is not reasonable to put much effort into baroreceptor modeling [2], but if all following model equations build on it, a more detailed approach is desirable at this point. Although baroreceptors can be seen as proportionally and differentially reacting controllers, other factors were ignored, such as adaption to changes in mean pressure, saturation, and neural conduction times. They could easily be included, as for example DeBoer et al. showed, using a scaled arctangent function to mimic saturation at high and low pressure levels [11].

For the third model, three different types of baroreceptors were presented, in order to account for the variation in thresholds for different baroreceptor types. They also include a maximum firing rate, which is a necessary assumption, since the firing rate is generally limited by the duration of the absolute refractory phase of baroreceptor ion channels. Their dependency on mean blood pressure over the last 1, 5, or 250 seconds gives them an adaptability to longer phases of hypertension, clearly missing in the first model[16]. Moreover, this modeling approach still presents an easily implementable, yet physiologically much more accurate basis for the input of the ANS, since at least two different types of baroreceptors are known [31].

ANS Activity The first model includes the basic characteristics of sympathetic and parasympathetic activity. They both show an oscillating behaviour and are modulated by baroreceptor activity, giving them a seemingly antagonistic behaviour. The basic frequency of both branches of the ANS was set equal to the respiratory frequency, which is contrary to the fact, that respiratory changes in HR were shown to be modulated mainly by parasympathetic activity [15]. The first model also modulates sympathetic activity proportional to the current baroreceptor activity, which is an assumption incompatible to the fact, that sympathetic activity is widely attributed to the low frequency components of HRV [28]. The erroneous regulation of sympathetic activity is also evident, if one considers that a maximum function had to be included into the model, in order to prevent negative sympathetic activity values. Another shortcoming lies in the complete absence of autonomous ANS activity independent from modulating factors.

Therefore, two oscillators for fundamental ANS activity, a low pass filter for sympathetic activity, and a term for respiration only in the parasympathetic nervous activity were included in the second model. The simulation results were superior to those of the first one by far, but still undesirable phases of zero sympathetic activity occurred, resulting in a lack of HRV. This leads to the conclusion, that baroreceptor activity and respiratory influences should not be included directly into sympathetic activity, if one locates sympathetic activity in the low frequency spectrum.

For the third model, the absence of independent ANS activity was analogously evened out by the inclusion of two oscillators. Since sympathetic activity was again directly dependent on the baroreceptor firing rate, another low pass filter was included. This causes a reaction to quick changes in blood pressure mainly mediated via the parasympathetic nervous system, which is consistent since this ANS part is attributed to the high frequency components of HRV.

Respiration Respiration, as a main influence on short term HRV, was included into the models in two different ways: The first and second model include respiration in the ANS, which is comprehensible, since changes in parasympathetic activity exist due to central respiratory modulation [15]. A disadvantage lies in the resulting temporal shift between inspiration and augmentation of the HR. The third model avoids this by varying the IPFM threshold based on the current level of in- and exhalation, leading to an instantaneous change of HR. It still is questionable, whether respiratory sinus arrhythmia should be modelled via the parasympathetic nervous system, or directly influence HR.

Blood Pressure Effective blood pressure regulation serves as an indicator of how well the model is constructed, since controlling mechanisms regulate the HR with the aim of keeping blood pressure levels within a normal range. In Studies, systolic blood pressure was shown to increase during phases of higher HR, whereas diastolic blood pressure was only slightly augmented in all age groups and for all heart rates [32]. The first and second model show significant changes in systolic, as well as diastolic blood pressure. Additionally, overestimated average blood pressure levels cause the previously mentioned phases of zero sympathetic activity. This is clearly not a desirable outcome and can be attributed to a combination of inadequacies. Quick changes in blood pressure during systole are normally evened out by the windkessel effect of the aorta. Although the first and second model do include a windkessel time constant, it only effects the diastolic pressure decrease. During systole, only the contractility varies based on sympathetic activity. This combination results in overly high blood pressure levels. Also, stroke volume and its dependence on the HR was not included, which further increases blood pressure levels. In comparison to the first two models, the third one showed a very stable mean systolic blood pressure for all patients and all mean heart rates, whereas the diastolic pressure showed fluctuations. The model expansion including HR dependent stroke volume and an arterial windkessel clearly had a stabilising effect on blood pressure. Especially the windkessel regulates blood pressure during systole, as well as diastole, and gives a more physiologic shape to the pressure curve. Although the third model behaves more realistical than the other two, correct selection of compliance and resistance is essential for an accurate calculation of blood pressure. Furthermore, fluctuations in rapid changes in diastolic blood pressure should be avoided generally. What is missing in the third model, is not only stroke volume, but also contractility depending on the filling of the heart. Varying peripheral resistance over time also presents a possibility for further model extension.

7 Conclusion

Although lots of existing models and submodels of the CVS include a variety of regulatory mechanisms of the HRV, they reveal shortcomings when closer examined. A lot of them do not address physiological processes in detail, or are largely based on purely statistical evaluations.

The first implemented model includes a variety of regulatory mechanisms, such as baroreceptors, sympathetic and parasympathetic activity, neurotransmitter concentrations, and myocardial contractility, but despite the many regulatory mechanisms it failed to mimic even the mean HR of the given subject data correctly.

The second model, which resulted from enhancements based on medical considerations of the first one, showed significant improvements in model performance when compared to subject data. However, it still generates physiologically untenable outputs for some cases. Phases of zero sympathetic activity caused by poor modeling of the baroreceptors, high fluctuations in systolic and diastolic blood pressure, and an incomplete arterial windkessel still provide starting points for further model enhancements.

The third model, being a combination of physiologically well-founded models, includes a more accurate description of the baroreceptors, and was also extended to further enhance its performance. Nevertheless, it overestimates short term as well as long term variability, which may be due to an inadequate choice of input for the IPFM model. In order to adequately recreate the mean HR given by the subject data, the IPFM input had to be evened out by a correction term analogously to the one used for the second model. In summary, models of cardiovascular regulation with an emphasis of correct represenation of HRV do exist, but only some of them are able to mimic reality to a satisfying degree. Modeling approaches, although physiologically valid, often put an emphasis on single aspects of the cardiovascular system, and simultaneously neglect others, which leads to a limited function of the whole model.

Once the existing regulatory mechanisms are accurately modelled, further enhancements may include age, gender, chemoreceptors, posture, hormonal regulation or even parts of the central nervous system. Nevertheless, further research needs to find a balance between a models richness of detail and an excessive amount of uncertain parameters.

References

[1] McCraty R, Shaffer F. Heart rate variability: new perspectives on physiological mechanisms, assessment of self-regulatory capacity, and health risk. *Global Advances in Health and Medicine*. 2015;4(1):46–61.

[2] Seidel H, Herzel H. Modelling heart rate variability due to respiration and baroreflex. In: *Modelling the Dynamics of Biological Systems*, pp. 205–229. 1995;.

[3] Acharya UR, Joseph KP, Kannathal N, Min LC, Suri JS. Heart rate variability. In: *Advances in cardiac signal processing*, pp. 121–165. 2007;.

[4] McCorry LK. Physiology of the autonomic nervous system. *American journal of pharmaceutical education*. 2007;71(4):78.

[5] Pape HC, Kurtz A, Silbernagl S. *Physiologie*. 2018.

[6] Guyton A, Hall J. Textbook of medical physiology, 11th. 2006.

[7] Yasuma F, Hayano Ji. Respiratory sinus arrhythmia: why does the heartbeat synchronize with respiratory rhythm? *Chest Journal*. 2004;125(2):683–690.

[8] Teich MC, Lowen SB, Jost BM, Vibe-Rheymer K, Heneghan C. Heart rate variability: measures and models. *arXiv preprint physics/0008016*. 2000;.

[9] Barbieri R, Matten EC, Alabi AA, Brown EN. A point-process model of human heartbeat intervals: new definitions of heart rate and heart rate variability. *American Journal of Physiology-Heart and Circulatory Physiology*. 2005;288(1):H424–H435.

[10] McLernon D, Dabanloo NJ, Ayatollahi A, Majd VJ, Zhang H. A new nonlinear model for generating RR tachograms. In: *Computers in Cardiology, 2004*. IEEE. 2004; pp. 481–484.

[11] DeBoer RW, Karemaker JM, Strackee J. Hemodynamic fluctuations and baroreflex sensitivity in humans: a beat-to-beat model. *American Journal of Physiology-Heart and Circulatory Physiology*. 1987; 253(3):H680–H689.

[12] Mateo J, Laguna P. Improved heart rate variability signal analysis from the beat occurrence times according to the IPFM model. *IEEE Transactions on Biomedical Engineering*. 2000;47(8):985–996.

[13] Bailón R, Laouini G, Grao C, Orini M, Laguna P, Meste O. The integral pulse frequency modulation model with time-varying threshold: application to heart rate variability analysis during exercise stress testing. *IEEE transactions on biomedical engineering*. 2011; 58(3):642–652.

[14] Brennan M, Palaniswami M, Kamen PW. A new cardiac nervous system model for heart rate variability analysis. In: *Engineering in Medicine and Biology Society, 1998. Proceedings of the 20th Annual International Conference of the IEEE*. IEEE. 1998; pp. 349–352.

[15] Eckberg DL. Point: counterpoint: respiratory sinus arrhythmia is due to a central mechanism vs. respiratory sinus arrhythmia is due to the baroreflex mechanism. *Journal of applied physiology*. 2009;106(5):1740–1742.

[16] Olufsen MS, Tran HT, Ottesen JT, Lipsitz LA, Novak V. Modeling baroreflex regulation of heart rate during orthostatic stress. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*. 2006;291(5):R1355–R1368.

[17] Ursino M. Interaction between carotid baroregulation and the pulsating heart: a mathematical model. *American Journal of Physiology-Heart and Circulatory Physiology*. 1998;275(5):H1733–H1747.

[18] Kamen PW, Krum H, Tonkin AM. Poincare plot of heart rate variability allows quantitative display of parasympathetic nervous activity in humans. *Clinical science*. 1996;91(2):201–208.

[19] Brennan M, Palaniswami M, Kamen P. Do existing measures of Poincare plot geometry reflect nonlinear features of heart rate variability? *IEEE transactions on biomedical engineering*. 2001;48(11):1342–1347.

[20] Bazan O, Ortiz JP. Duration of Systole and Diastole for Hydrodynamic Testing of Prosthetic Heart Valves: Comparison Between ISO 5840 Standards and in vivo Studies. *Brazilian journal of cardiovascular surgery*. 2016;31(2):171–173.

[21] Aysin B, Aysin E. Effect of respiration in heart rate variability (HRV) analysis. In: *Engineering in medicine and biology society, 2006. EMBS'06. 28th annual international conference of the IEEE*. IEEE. 2006; pp. 1776–1779.

[22] Westerhof N, Lankhaar JW, Westerhof BE. The arterial windkessel. *Medical & biological engineering & computing*. 2009;47(2):131–141.

[23] Parragh S. Modelle zur Bestimmung des aortalen Blutflusses basierend auf Optimalitätsbedingungen. Master's thesis, TU Wien, Institut für Analysis und Scientific Computing. 2013.

[24] Boudoulas H, Geleris P, Lewis RP, Rittgers SE. Linear relationship between electrical systole, mechanical systole, and heart rate. *Chest*. 1981;80(5):613–617.

[25] Van de Vooren H, Gademan MG, Swenne CA, TenVoorde BJ, Schalij MJ, Van der Wall EE. Baroreflex sensitivity, blood pressure buffering, and resonance: what are the links? Computer simulation of healthy subjects and heart failure patients. *Journal of Applied Physiology*. 2007;102(4):1348–1356.

[26] Barbi M, Chillemi S, Garbo AD, Balocchi R, Menicucci D. A minimal model for the respiratory sinus arrhythmia. *Biological cybernetics*. 2006; 94(3):225–232.

[27] Bachler M, Sehnert W, Mikisek I, Mengden T, Wassertheurer S. Abstract 15861: Device-Guided Slow Breathing With Real-Time Biofeedback for Personalized Cardiovascular Therapy Support Increases Pulse Transit Time. *Circulation*. 2018;Vol 138.

[28] Malik M. Heart rate variability. *Annals of Noninvasive Electrocardiology*. 1996;1(2):151–181.

[29] Meste O, Khaddoumi B, Blain G, Bermon S. Time-varying analysis methods and models for the respiratory and cardiac system coupling in graded exercise. *IEEE transactions on biomedical engineering*. 2005;52(11):1921–1930.

[30] Antelmi I, De Paula RS, Shinzato AR, Peres CA, Mansur AJ, Grupi CJ. Influence of age, gender, body mass index, and functional capacity on heart rate variability in a cohort of subjects without heart disease. *The American journal of cardiology*. 2004; 93(3):381–385.

[31] Lohmeier TE, Iliescu R. The baroreflex as a long-term controller of arterial pressure. *Physiology*. 2015; 30(2):148–158.

[32] Miyai N, Arita M, Miyashita K, Morioka I, Shiraishi T, Nishio I. Blood pressure response to heart rate during exercise test and risk of future hypertension. *Hypertension*. 2002;39(3):761–766.

Solving ARGESIM Benchmark C22 'Non-standard Queuing Policies' with MatlabGPSS

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(4), 2019, 199-205, DOI: 10.11128/sne.29.bn22.10496
Received: November 18, 2019;
Accepted: November 22, 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' deals with queueing systems, where entities can leave a queue in a dynamically changed order. It is solved here with MatlabGPSS, a Matlab-based implementation of the well-known GPSS modeling language. Though quite old, it still has its merits, such as precisely defined and flexible methods to deal with concurrent events and the concept of *user chains*, which are more flexible than standard queueing blocks.

Introduction

The ARGESIM benchmark C22 [1] is concerned with special queueing systems, which use a FIFO discipline generally, but where some entities can leave the queue prematurely or in a dynamically changed order. Its implementation can be difficult in modeling environments, which provide only monolithic queueing blocks that don't grant access to the entities waiting in a queue [2].

GPSS [3] is one of the oldest existing modeling languages. It uses the transaction-based paradigm to model discrete event systems, and though being somewhat outdated, it is still a good example of a simple language that uses only a few basic constructions to provide very wide modeling capabilites. Therefore it should be a good tool to study principal questions such as appropriate concepts for modeling of non-standard queues.

Among the few still existing implementations we chose MatlabGPSS [4], because it is easily available [5] and combines GPSS statements with general Matlab code. This makes the implementation of complex control structures and the compilation of statistical and graphical results much easier than relying on pure GPSS constructs, thereby allowing to concentrate on the basic questions of queue design.

GPSS is text-based and contains statements for the generation and destruction of entities (`generate`/`terminate`), the entering and leaving of queues (`queue`/`depart`) and the reservation and freeing of servers (`seize`/`release`). The `advance` statement delays an entity for a given time and is used to model service times. Each entity can store an arbitrary number of parameters `P(i)`, similar to attributes in other modeling environments. Additional functions `Q(i)`, `F(i)` and `CH(i)` return the number of entities stored in the i-th queue, server (*facility*) or user chain.

Despite its name the `queue` statement does not implement any queueing discipline, it is only used for accounting purposes, e.g. to count the number of entities entering a queue. Usually a simple FIFO discipline within a priority level of entities is defined automatically by the internal scheduling of the GPSS system. For more complex queueing strategies one can use so called *user chains*, which are more flexible than standard queues. Entities join a user chain with the `link` statement, entering at the front or end or according to a parameter value. The `unlink` statement allows any entitiy to free one or more entities from a user chain and to route them to arbitrary places. The exact possibilities of `unlink` depend on the specific GPSS implementation; in MatlabGPSS entities can only be extracted from the front or back end of a user chain.

In the following we will implement all tasks of the C22 benchmark, with a special focus on the advantages and shortcomings of the queueing mechanisms in MatlabGPSS and the available methods to specify the ordering of concurrent events. For the exact definition of all systems and parameters refer to the benchmark definition [1]. The models and scripts necessary to reproduce all results presented here are available from [6].

1 Basic Queuing System

1.1 Description of the Basic Model

The basic model can be easily implemented in GPSS using standard methods. The complete source code – omitting most variable initialisations and code for statistical results and plots, and adding line numbers for easier reference – is given by:

```
 1  init
 2    nQ = 4;
 3  model
 4    generate(tA,0,1,nE,0)
 5    [~, nr] = min(Q(1:nQ) + F(1:nQ));
 6    P(3, nr);
 7    queue(P(3),1)
 8    seize(P(3))
 9    depart(P(3),1)
10    advance(tS,0)
11    release(P(3))
12    terminate(1)
```

The only interesting part is in line 5 and 6, where the number of the shortest queue (including server allocation) is computed and stored in an entity parameter.

1.2 Results of Deterministic Model

Entity ids are stored in a parameter, the queue lengths are collected using GPSS function `Q` and plotted with standard Matlab methods. The plots of the ids of the last 20 outgoing entities over their exit time and of the total queue length are shown in Figure 1.

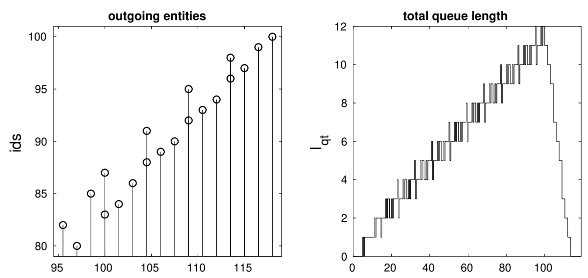

Figure 1: Outgoing ids and total queue length for the basic model.

1.3 Results of Stochastic Model

The stochastic model uses MatlabGPSS functions `expod` and `triangd` to create exponentially or triangularly distributed random numbers. For the computation of waiting times the entry times are stored in an entity parameter. Table 1 displays statistical results for the total queue length and entity waiting times.

l_{qt}		$t_{q,i}$	
avg	**max**	**avg**	**max**
21.131	45.000	24.075	52.696

Table 1: Total queue lengths and queue waiting times for the basic model.

1.4 Variants for Handling Simultaneous Events

If several entities can move at the same time, their order can be defined in GPSS using the command `priority N`, which assigns an integer priority $N \geq 0$ to an entity, where entities with larger priorities take precedence over those with smaller priorities. The concurrency order of entities with the same priority is not defined in general, but depends on the GPSS implementation: While in GPSS/H the order is specified, in GPSS World it is explicitely randomised [7]. Therefore one should not rely on any given order, but use priorities to guarantee the required behaviour.

For the basic model there are two critical situations to consider:

1. An entity E_1 wants to leave a server at the same time that a queued entity E_2 wants to enter a server or a new entity E_3 is generated.

2. A queued entity E_2 wants to enter a server at the same time that a new entity E_3 is generated.

To get the required ordering E_1 needs a higher priority than E_2, which needs a higher priority than E_3. This can easily be achieved by generating events with priority 0 and inserting the statements `priority 1` between lines 7 and 8 of the source code above and `priority 2` between lines 9 and 10. Using knowledge about the internal scheduling algorithm one could reduce this to one priority statement.

To implement the alternative order, where a new entity joins a queue, *before* another one leaves the server, one generates all entities with priority 1 and lowers the priority to 0 before an entity releases its server, e. g. between lines 9 and 10 of the source code above. This works, since the second critical situation is not possible here. The result is shown in Figure 2.

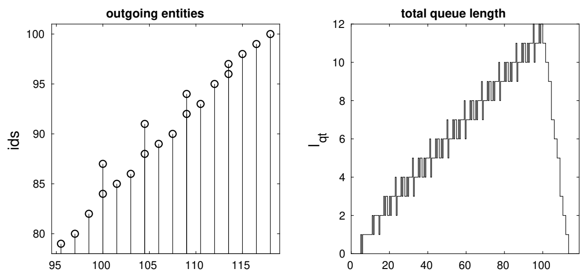

Figure 2: Outgoing ids and total queue length for the concurrency variant.

1.5 Comparison of Standard and Large Version

The implementation of the large model in MatlabGPSS is trivial: One only has to change the parameters, especially $n_Q = 40$, and gets the results shown in Table 2.

$\mathbf{l_{qt}}$		$\mathbf{t_{q,i}}$	
avg	**max**	**avg**	**max**
241.598	488.000	27.706	62.494

Table 2: Total queue lengths and queue waiting times for the large model.

2 Jockeying Queues

2.1 Description of the Model

To implement the jockey queue system, one has to move entities between different places in the model. This is done with the GPSS statement `transfer` and labelling of statements, similar to a classical *go-to*. Since such source code is hard to read, it is represented here as a flow chart in Figure 3. In a graphical modeling environment similar models would use routing elements such as gates and switches.

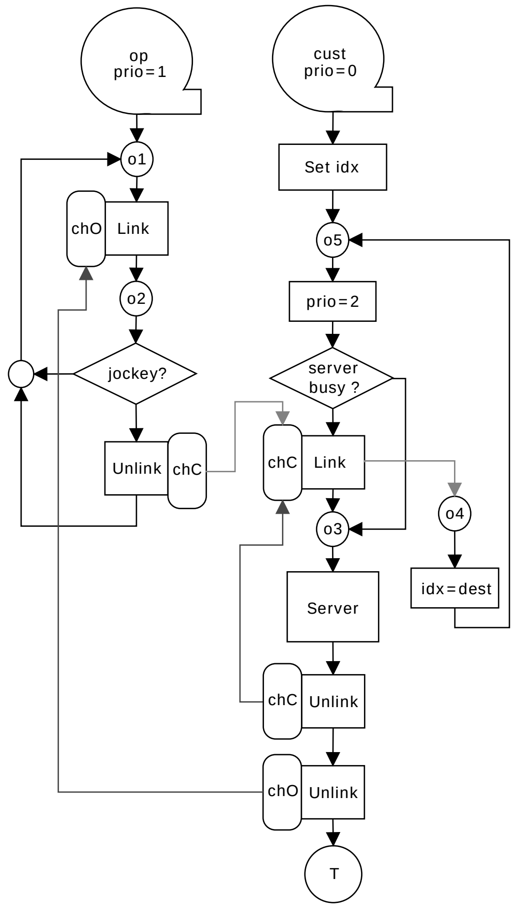

Figure 3: Flowchart of the jockey queues model.

The left side shows the creation of a single entity `op` that acts as an operator: It checks, whether jockeying is possible and moves the corresponding entity from the end of its queue to the end of the shorter one. Since jockeying can only happen when a queue gets shorter after service completion, the operator entity waits in a user chain `chO`, until it is unlinked by a leaving entity. The actual computation of source and destination queues is done with a simple Matlab function.

The right side describes the life of a normal customer entity: After creation it computes the number `idx` of the shortest queue available, then it enters the user chain `chC(idx)`, if the server is busy. It is unlinked either from the head of the queue by a served entity to get to the server next or from the end by the operator to jockey to a different queue.

To make sure that jockeying happens before the generation of a new entity, the operator is generated with priority 1, while the customer entities start with priority 0. Their priority is raised to 2 before they enter a chain or a server, so that servicing and moving up have precedence over jockeying.

2.2 Results of Deterministic Model

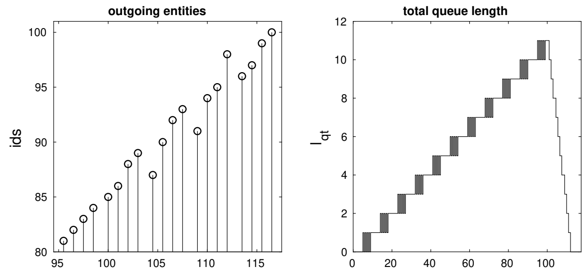

Figure 4: Outgoing ids and total queue length for the jockeying model.

The requested plots are shown in Figure 4 and the first five and last five jockeying events in Table 3.

t	id	src	dest
6.5	6	1	2
7.5	7	1	3
8.5	8	1	4
11.0	10	1	2
12.0	11	1	3
..	..	..	..
89.5	89	2	4
93.0	92	2	3
94.0	93	2	4
98.5	98	3	4
103.0	100	1	4

Table 3: First and last five jockeying events.

2.3 Results of Stochastic Model

To compute total queue waiting times for jockeying entities, the single queue waiting times are accumulated in entity parameters. The statistical results are displayed in Table 4. The number of jockeying events in the corresponding run was 131.

l_{qt}		$t_{q,i}$	
avg	**max**	**avg**	**max**
20.842	45.000	23.618	51.408

Table 4: Total queue lengths and queue waiting times for the jockey model.

3 Reneging Queues

3.1 Description of the Model

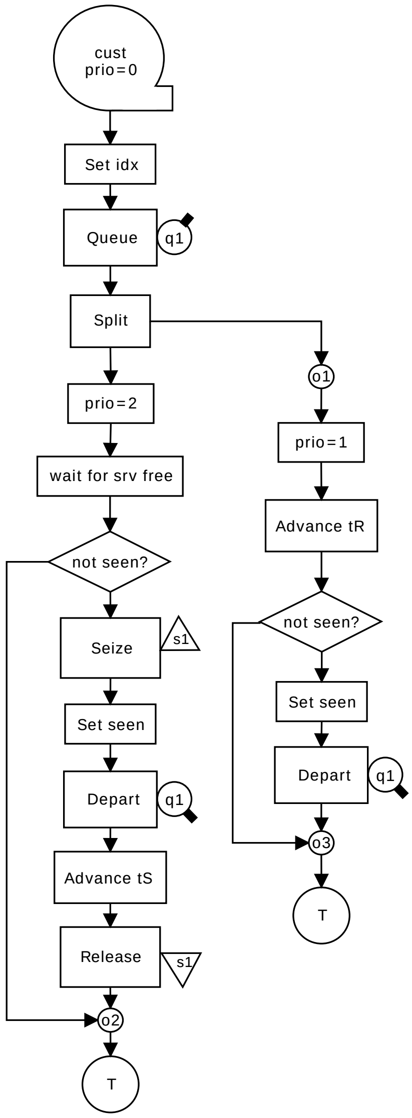

Figure 5: Flowchart of the renege queues model.

The chosen implementation of the reneging queue is similar to the *clone queue* presented in [2]. It uses the GPSS statements `split` to create a copy of an entity and `gate_fnu()`, which halts an entity until a given server is free. The basic idea is shown in Figure 5: After entering a queue an entity is cloned, one copy waits for the total reneging time `tR`, the other one tries to get the server. A bookkeeping variable `seen` is set, whenever one of the pair is ready, and checked before the action of the other one. A clone that comes late, is simply terminated.

To obtain the required order of concurrent events, entities are generated with priority 0. The priority is raised to 1 for the clones that are going to renege, and to 2 for those that wait for the server.

3.2 Results of Deterministic Model

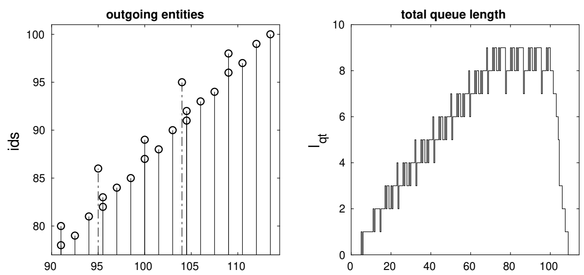

Figure 6: Outgoing ids and total queue length for the reneging model.

Figure 6 shows the plots of the ids of the last outgoing entities including reneging entities and the total queue length. The reneging events are listed in Table 5.

t	id
77.0	68
86.0	77
95.0	86
104.0	95

Table 5: Reneging events.

3.3 Results of Stochastic Model

The statistical results are displayed in Table 6. The number of reneging events in the corresponding run was 53.

l_{qt}		$t_{q,i}$	
avg	max	avg	max
4.615	15.000	4.882	9.000

Table 6: Total queue lengths and queue waiting times for the renege model.

4 Classing Queues

4.1 Description of the Model

The implementation of the classing queue uses a single entity for the operator and two user chains for each classing queue: `c1` for incoming entities and `c2` for the entities, whose class is currently being served (cf. Figure 7). New GPSS constructs used here are logical variables that work like gates in graphical environments, together with the functions `logic_s()`/`logic_r()` to set/reset a variable and `gate_ls()` to wait until a variable is set.

After an initial waiting time t_C the operator entity checks whether there are any entities to be called, if necessary it waits until entities arrive. Then it computes the next active class and unlinks all customer entities from the chain `c1`. After the customer entities of the current class have been served, the operator loops to call the next class.

New customer entities first enter the `c1` chain, choosing the shortest queue as always. When the operator calls for a new class, the matching entities proceed to their `c2` chain and wait to be served, the other ones reenter `c1`. This behaviour is similar to the shuffle queue pattern in [2]. Before leaving the system after service, a customer entity checks, whether there are still entities of its class being served, and wakes up the operator, if not.

Due to the complex “jockeying” structure of the class queue one needs a bit of fine-tuning to obtain the requested order of concurrent events: The operator is created with priority 0, the customer entities with priority 1. This guarantees that the operator call always comes last among concurrent events.

When an entity enters the queue, its priority is risen to 3, so that entering and leaving of the server will preceed the arrival of a new entity. After the entity has left the server and freed the next entity from its chain, its priority is lowered to 2 so that its successor can enter the server, before the server state is checked.

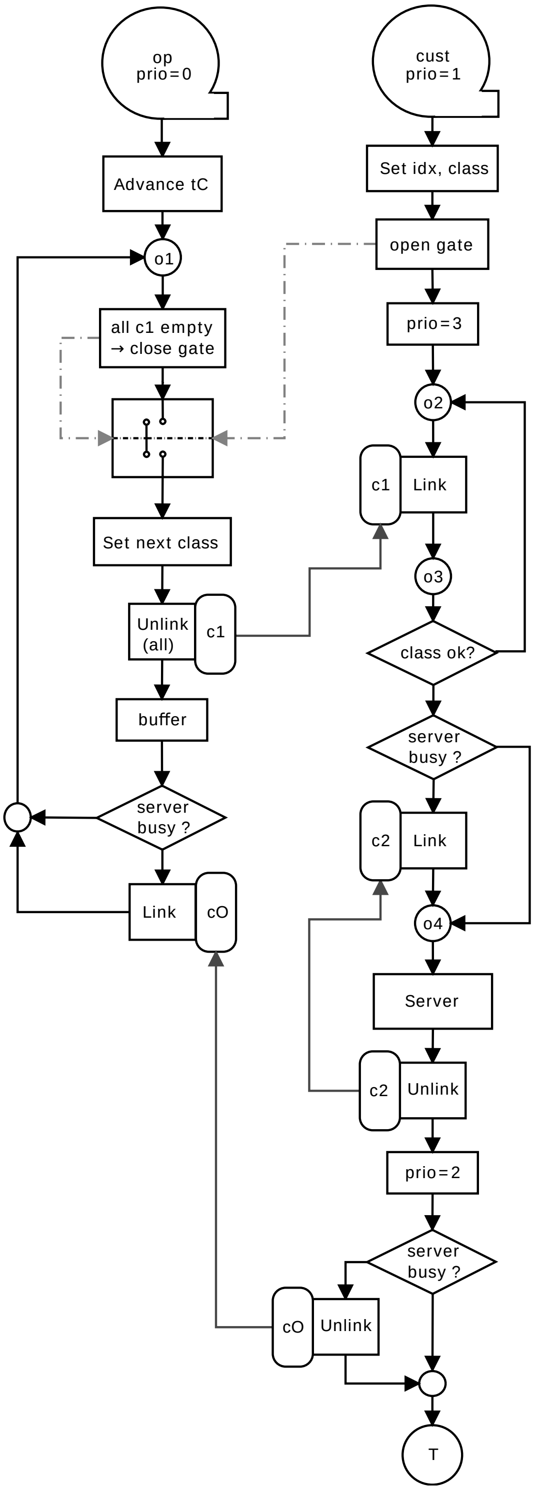

Figure 7: Flowchart of the class queues model.

After the operator has unlinked all entities in the `c1` chains, it has to wait, until any of the freed entities have claimed a server, before checking. This is done with the GPSS `buffer` statement that gives up immediate control and moves the calling entity to the end of the chain of concurrent events.

4.2 Results of Deterministic Model

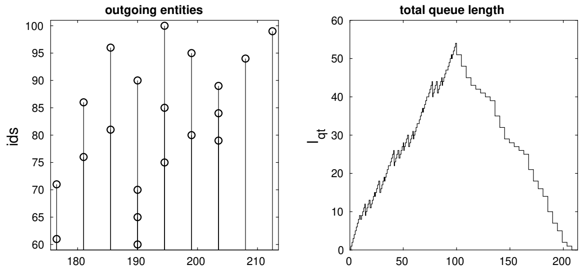

Figure 8: Outgoing ids and total queue length for the classing model.

The plots of the ids of the last outgoing entities and the total queue length are shown in Figure 8, average and maximal queue waiting times per class in Table 7.

class	1	2	3	4	5
avg	66.80	60.85	41.85	50.98	60.77
max	121.50	104.00	77.00	120.00	125.50

Table 7: Queue waiting times per class (deterministic model).

4.3 Results of Stochastic Model

The usual statistical results are displayed in Table 8, average and maximal queue waiting times per class in Table 9.

$\mathbf{l_{qt}}$		$\mathbf{t_{q,i}}$	
avg	**max**	**avg**	**max**
82.615	152.000	122.286	279.325

Table 8: Total queue lengths and queue waiting times for the class model.

class	1	2	3	4	5
avg	130.99	118.26	109.18	124.43	126.93
max	279.32	255.62	240.06	273.24	272.97

Table 9: Queue waiting times per class (stochastic model).

5 Conclusions

The combination of basic, but flexible GPSS statements with the versatility of the Matlab environment has made possible a straightforward implementation of all benchmark tasks. Matlab has been especially useful for the overall control structure of the programs, for computations like the determination of source and destination in the jockeying case and for calculating and plotting results.

The immediate advantages of a textual modeling language as compared to a graphical environment have been shown by the trivial implementation of a system with 40 queues. Another bonus is the handling of concurrent events: The behaviour can be flexibly adapted using priorities and the `buffer` statement.

However, the central point of this investigation, namely the flexibility of queue modeling constructs, has shown mixed results: On the one hand the separation of queue storage (`link`) and queue control (`unlink`) with different entities simplified the control structure of the models, and the possibility to extract entities from the front and the back end of a queue allowed for a simple implementation of the jockey queue. On the other hand, for the renege and class queues we had to use clumsy constructs like the clone and shuffle queues, because MatlabGPSS provides no direct access to entities within a queue. Here other GPSS implementations had more options, e. g. the extraction of entities using boolean expressions containing entity parameters [3].

In view of the age of MatlabGPSS an appropriate extension seems to be pointless. Nevertheless the flexibility and preciseness of GPSS are still outstanding features. Whether modern graphical environments can compete, will hopefully be seen by future implementations of the C22 benchmark.

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] Junglas P, Pawletta Th. Non-standard Queuing Policies: Definition of ARGESIM Benchmark C22. *Simulation Notes Europe SNE*. 2019; 29(3): 111-115. doi: 10.11128/sne.29.bn22.10481

[2] 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

[3] Schriber Th J. *An introduction to simulation using GPSS/H*. New York: John Wiley & Sons, Inc.; 1991. 437 p.

[4] Pawletta Th, Drewelow W, Pawletta S. Discrete Event Simulation in Interactive Scientific and Technical Computing Environments. In: *Proc. 12th European Simulation Multiconference on Simulation*; 1998 Jun; Manchester. 529-533. ISBN 1-56555-148-6.

[5] Pawletta Th., et al. *The MATLAB GPSS Toolbox*. Online: `http://www.cea-wismar.de/tbx/mgpss/` (called 2019-11-14).

[6] Junglas P. Argesim C22 models and scripts. Online: `http://www.peter-junglas.de/fh/simulation/argesimc22.html` (called 2019-11-14).

[7] Minuteman Software. *GPSS World Reference Manual*. Online: `http://minutemansoftware.com/reference/reference_manual.htm` (called 2019-11-14).

Python-based eSES/MB Framework: Model Specification and Automatic Model Generation for Multiple Simulators

Hendrik Folkerts[1*], Thorsten Pawletta[1], Christina Deatcu[1], Sven Hartmann[2]

[1]Research Group Computational Engineering and Automation, University of Applied Sciences Wismar, Philipp-Müller-Straße 14, 23966 Wismar, Germany; *hendrik.folkerts@cea-wismar.de

[2]Department of Informatics, Clausthal University of Technology, Julius-Albert-Straße 4, 38678 Clausthal-Zellerfeld, Germany

SNE 29(4), 2019, 207-215, DOI: 10.11128/sne.29.tn.10497
Received: July 20, 2019 (Selected EUROSIM 2019 Postconf. Publ.), Accepted: September 22, 2019
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. This paper proposes a Python-based infrastructure for studying the characteristics and behavior of families of systems. The infrastructure allows automatic execution of simulation experiments with varying system structures as well as with varying parameter sets in different simulators. Possible system structures and parameterizations are defined using a System Entity Structure (SES). The SES is a high level approach for variability modeling, particularly in simulation engineering. An SES describes a set of system configurations, i.e. different system structures and parameter settings of system components. In combination with a Model Base (MB), executable models can be generated from an SES. Based on an extended SES/MB approach, an enhanced software framework is introduced that supports variability modeling and automatic model generation for different simulation environments. By means of an engineering application it is shown, how a set of Python-based open source software tools can be used to model an SES and to automatically generate and execute signal-flow oriented models.

Introduction

This paper is a modified version of [1]. It clarifies the scope of the discussed tools for variability modeling.

Generally, variability modeling can be seen as an approach to describe more than one system configuration. A system configuration incorporates the structure of the model as well as the parameter settings. Different system configurations arise e.g. when modeling varying real world systems or when modeling system variations for finding an optimal system design.

In software engineering, a classical approach to variability modeling is the use of Feature Models (FM) in combination with 150% models. Feature Models specify components and relations, which are used for modeling the variability of a system or product [2]. By selecting features a Variant Model (VM) can be generated from the FM. Thus, the VM represents a subset of all variants defined in the FM. The FM and the VM are platform-independent. For generating executable models, platform dependent dynamic models are needed, which are linked to the FM. All dynamic models are organized as an 150% model. With the help of a variant generator an executable model is generated based on a VM and the 150% model. The software pure::variants is a prominent example supporting this approach [3]. Another approach is the executable Unified Modeling Language with its standard Foundational Unified Modeling Language (fUML). The fUML is a subset of the graphical notation of the UML with executable semantics in order to generate platform dependent software [4]. Based on the UML the executable Systems Modeling Language (SysML) was developed in the field of systems engineering.

Analogous to approaches coming from software engineering the systems theory community introduced methods for platform-independent variability modeling with subsequent platform-dependent model generation of specific variants. One method is the System Entity Structure (SES) and Model Base (MB) approach. In this paper the theory of an extended SES/MB approach is discussed and software tools implementing the theory are presented using an engineering application example.

1 SES/MB Theory and Implementation

This section briefly discusses the general SES/MB theory and the derived extended SES/MB (eSES/MB) infrastructure. Subsequently, an implementation of the infrastructure is presented.

1.1 SES/MB Basics and the eSES/MB Infrastructure

An SES is represented by a tree structure comprising entity nodes, descriptive nodes and attributes. Different system structures can be coded in an SES tree. In the context of modeling and simulation *entity nodes* are linked to *basic models* organized in an MB. Attributes of an entity node correspond to the parameters of the associated basic model. *Descriptive nodes* describe the relations among at least two entities and are divided into *aspect*, *multi-aspect* and *specialization* nodes. An aspect node represents the composition of an entity. The multi-aspect node is a special aspect node and describes the decomposition of an entity into several entities of the same type. The specialization node describes the taxonomy of an entity. Descriptive nodes can specify variation points using specific rules.

Specifying a composition of entities, which describes the composition of models in a modular-hierarchical manner, requires a specification of *coupling* relations. Couplings can be specified as properties of descriptive nodes of the type aspect or multi-aspect and consist of pairs of entity names and port names, such as *(source entity, source port, sink entity, sink port)*.

In order to derive a specific system configuration all variation points are resolved by evaluating the rules at the descriptive nodes of the SES. This procedure is called *pruning*. If more than one aspect or multi-aspect node appear on the same tree level – so called siblings –, they form a variation point with an xor-selection. A second kind of variation point for an xor-selection is defined by specialization nodes. When selecting one child entity of the specialization, the name and the properties of the selected child entity are merged with the name and properties of the father entity of the specialization. This procedure is formalized by the SES inheritance axiom. The resulting *Pruned Entity Structure (PES)* represents exactly one system configuration. In conjunction with an MB, a fully configured and executable model can be generated from the PES.

The basic SES/MB framework introduced in [5] was extended in [6] and [7] by new modeling features, methods and components, such as an *Experiment Control (EC)* and an *Execution Unit (EU)* as shown in Figure 1. In this eSES/MB infrastructure, the EC uses an interface to the SES and its methods to derive goal-driven system configurations and to generate models, which are executed by the EU. The results returned by the EU are collected and analyzed by the EC. Thus, the derivation and generation of subsequent system configurations can be controlled reactively based on experiments already carried out.

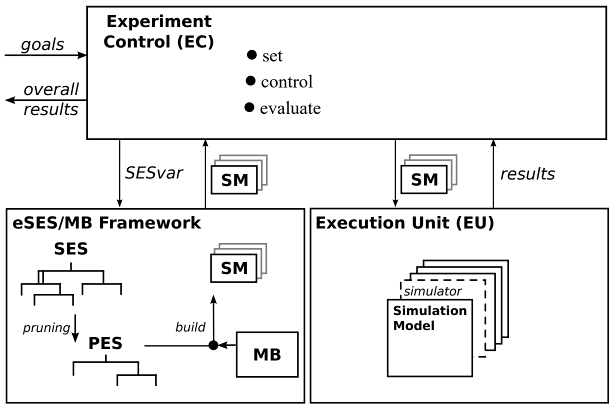

Figure 1: The eSES/MB infrastructure.

A set of variables with global scope establish the interface to the SES. They are called *SES variables* (SESvar). *Semantic conditions* can be used to specify permitted value ranges and dependencies between SESvars. *SES functions* (SESfcn) are introduced for the specification of procedural knowledge. Complex variability can often be described more easily with SESfcns. Typical examples include the definition of varying coupling relations or the definition of variable parameter configurations in attributes. For automatic pruning, *selection rules* at descriptive nodes need to be defined, such as *aspectrules* for aspect and multi-aspect siblings or *specrules* at specialization nodes. A special mandatory attribute of multi-aspects is the attribute *number of replications* (numRep). The numRep attribute specifies the number of entities to create at a multi-aspect node during pruning. The *mb-attribute* of leaf entity nodes connects the entity node to a basic model in the MB. Attribute values and selection rules can be specified using SESvars or SESfcns.

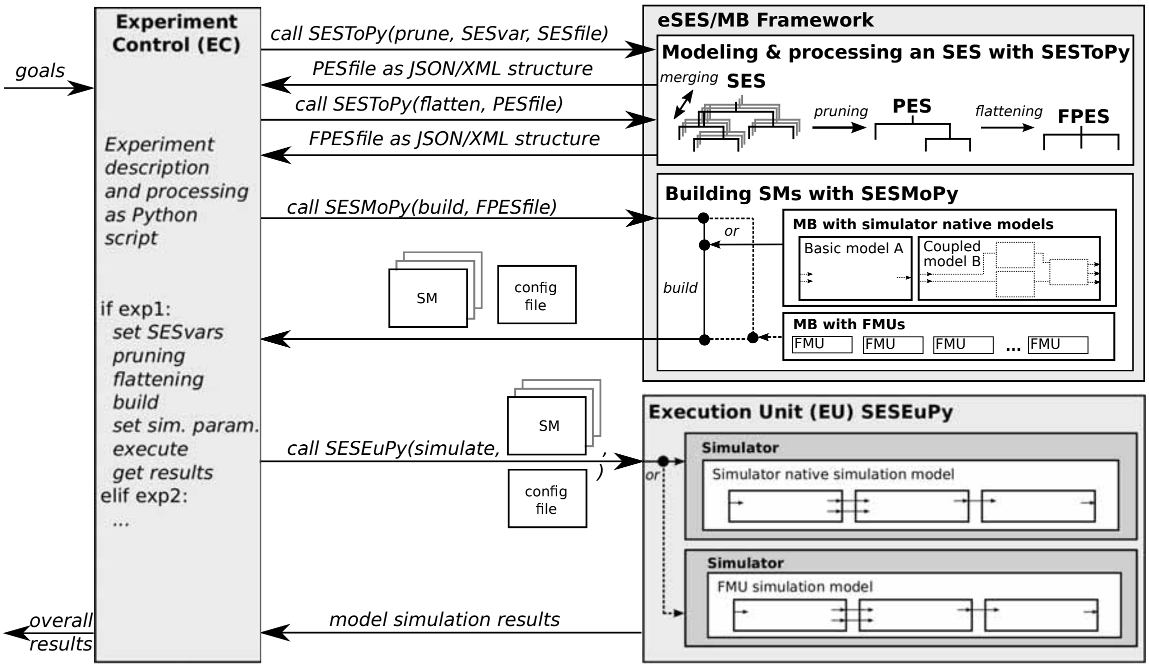

Figure 2: Python-based eSES/MB infrastructure for multiple EUs.

1.2 Software Tools

The eSES/MB framework as presented in the lower left part of Figure 1 was implemented in a prototype software tool in MATLAB [8]. The focus of this tool is the modeling and generation of MATLAB/Simulink models. In contrast to the MATLAB prototype, the objective of the new prototype discussed in this paper is to support the generation and execution of models for different simulation environments. The infrastructure in Figure 1 is implemented as a Python framework as presented in Figure 2. The tools are called *SESToPy*, *SESMoPy*, and *SESEuPy* [9].

SESToPy (**S**ystem **E**ntity **S**tructure **To**ols **Py**thon) implements a graphical editor and all SES related methods. In the editor an SES tree can be specified interactively in a file browser view and attributes and rules can be defined for every node. In addition to the pruning method already mentioned, SESToPy supports some more methods such as *merging* different SES and *flattening* for removing the hierarchy information. Applying the flattening method, a *Flattened Pruned Entity Structure (FPES)* is derived. The pruning and flattening methods can be used interactively or as API methods. SESToPy supports to save SES, PES, and FPES as JSON or XML files.

For generating executable models, SESMoPy (**S**ystem **E**ntity **S**tructure **Mo**del builder **Py**thon) was developed. SESMoPy is a model builder, which implements the *build* method in two different ways and supports several simulation environments. For both approaches, all corresponding basic models must be organized in an MB, as shown in Figure 2. The first approach, called *native model generation*, is the generation of executables for a specific EU. Currently native models for *MATLAB/Simulink, Dymola, OpenModelica*, the *PDEVS for MATLAB* toolbox and the Python-based environment *DEVSimPy* [10] can be generated. Currently for MATLAB/Simulink, Dymola, and OpenModelica SESMoPy only supports the build method for the generation of signal-flow oriented models. Support for the generation of physical models is under development. The second approach is the *model generation based on the Functional Mock-up Interface (FMI)*. The FMI standard is defined for several uses, one of which is FMI for Model Exchange. The generalized interface FMI is supported by a number of established simulators [11]. Using this approach, SESMoPy builds a Functional Mock-up Unit (FMU) model that implements an FMI [12]. This approach is under development. Depending on the way of model generation, the processing in the corresponding EU is different.

SESMoPy's build method generates executable simulation models (SM) using the information in the FPES and basic models from a target specific MB. Information about the way the model is created can be provided in the EC calling SESMoPy or at the SES level according to the SES enhancements in [7].

The Python software tool SESEuPy (**S**ystem **E**ntity **S**tructure **E**xecution **u**nit **Py**thon) acts as a general EU. It implements a kind of wrapper for the integration of different simulation environments into the framework. The modular structure and well-defined interfaces allow to integrate even components of non-simulation-specific environments into the framework.

In the next section, the components and functionality of the Python framework are explained using the example of an engineering application.

2 Engineering Application

A feedback control system can be modeled using transfer functions describing the behavior of the components in frequency domain. Controlled variables in a feedback control system are usually influenced by disturbances. A common approach for minimizing the influence of predictable disturbances is adding a feedforward control. The system can be mapped to a signal-flow oriented model. In the following paragraphs it is described how the eSES/MB infrastructure can be used to design and test such a system using the introduced tools and native model generation for OpenModelica.

2.1 Problem Description

A process unit with a *PT1* behavior shall be controlled using a PID controller. A disturbance with a *PT1* behavior affects the output of the process unit. Different configurations of the PID controller shall be tested. If a defined regulatory goal is met, the current configuration of the PID controller is taken. Otherwise the structure is varied by adding a feedforward control to the system and different configurations of the PID controller are analyzed again. Figure 3 depicts a schematic representation of the application.

The system's behavior follows the *PT1* transfer function in Equation 1 and the step-shaped disturbance affects the output of the process unit with a *PT1* behavior according to Equation 2. The optional feedforward control is realized by subtracting the disturbing signal calculated by Equation 3 from the manipulated variable.

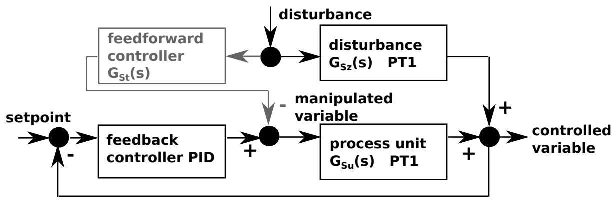

Figure 3: Structure of the feedback control system with optional feedforward control.

The control goals are a settling time of less than 15 seconds and a maximum overshoot of less than 5% after a disturbance.

The system has two structure variants, either without or with the feedforward control part, and a range of different configurations for the PID controller can be applied for each structure variant. In the next section, the two structure variants and their possible configurations are specified as an SES.

$$G_{Su}(s) = \frac{1}{20 \cdot s + 1} \tag{1}$$

$$G_{Sz}(s) = \frac{1}{10 \cdot s + 1} \tag{2}$$

$$G_{St}(s) = \frac{G_{Sz}(s)}{G_{Su}(s)} = \frac{20 \cdot s + 1}{10 \cdot s + 1} \tag{3}$$

2.2 Variant Modeling with SESToPy

The specification of the SES describing the feedback control system is done with the tool SESToPy. The tree and all attributes are defined via a graphical user interface. During modeling the SES with SESToPy, checks on the SES and plausibility tests are executed indicating model errors. The SES is saved as a JSON structure.

Figure 4 depicts the SES and its representation in SESToPy. The SES uses some extensions introduced in [13]. In addition to the different system configurations, essential parts for the configuration of simulation experiments are defined.

The root node *exp* of the SES and its subsequent aspect node *expDEC* describe a set of simulation based parameter studies for different system structures. The subtree of the entity node *simModel-ctrlSys* specifies the two system structures, i.e. a variant with and a variant without feedforward controller. The other two entity nodes specify experiment related information: The entity node *simMethod* specifies a target simulation

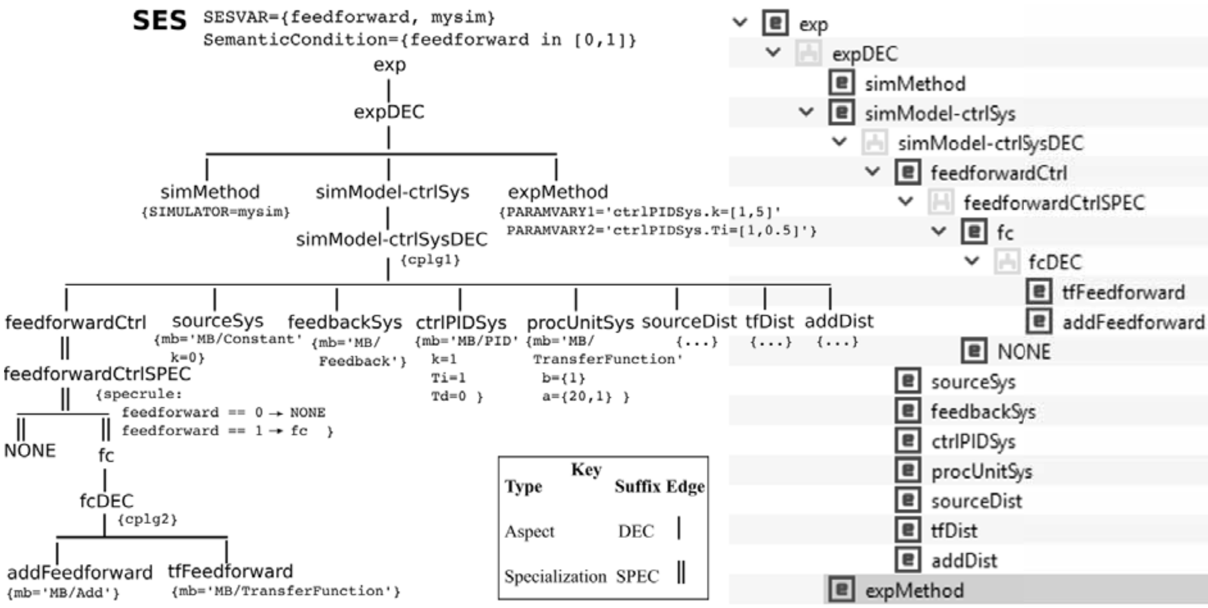

Figure 4: Left: SES specifying the feedback control system study; Right: Part of the SES representation in SESToPy.

environment for performing simulation runs using the SESvar *mysim*. Other simulation execution parameters, such as the simulation period, are not specified and are set by the EC later. The entity node *expMethod* specifies the permitted value ranges of two parameters for the PID controller. Besides the different system structures, they are the subject under study.

The aspect *simModel-ctrlSysDEC* describes that each system variant consists of the following entities: *sourceSys*, *feedbackSys*, *ctrlPIDSys*, *procUnitSys*, *sourceDist*, *tfDist* and *addDist*. They are mandatory system elements. The optional feedforward control is specified by the subtree of entity *feedforwardCtrl*. The coupling relations of both structure variants are defined in the attribute *cplg1* of aspect *simModel-ctrlSysDEC*, as discussed later.

According to [14], optional parts in an SES are expressed by a specialization node where one of its children is a NONE element. A NONE element means that the entity is not included at all. The selection at a specialization is defined by an attribute called *specrule*. The specrule of the specialization *feedforwardCtrlSpec* defines that either the entity *fc* or *NONE* is selected during pruning. The result of evaluating the specrule at node *feedforwardCtrlSPEC* depends on the value of the SESvar *feedforward*. The SESvar codes the two possible structure variants as values 1 or 0. Therefore, the semantic condition $feedforward \in [0,1]$ applies to the SESvar. The entity *fc* and its subsequent aspect *fcDEC* specifies the feedforward control structure as a composition of the two entities *tfFeedforward* and *addFeedforward*.

Aspects and multi-aspects can define coupling relations as attribute. These attributes are abbreviated with *cplg* in Figure 4. Due to the varying system structures specified in the SES, the couplings in attribute *cplg1* of aspect node *simModel-ctrlSysDEC* are defined using an SESfcn. The following code is an excerpt of the SESfcn.

```
def cplgfcn(feedforward):
  cplg = []

  #fixed couplings
  cplg.append(["sourceSys","y",
       "feedbackSys","u1",""])
  cplg.append([...

  #variable couplings
  if feedforward==0:
    cplg.append([...
  elif feedforward==1:
    cplg.append([...

  #return
  return cplg
```

The coupling definitions in *cplg2* at node *fcDEC* are invariable and can therefore be defined without using an SESfcn.

According to Section 1, each leaf node defines an mb-attribute referring to a basic model in the MB. The other attributes of the leaf nodes define properties to configure the linked basic models. The values for k and Ti specified at node *ctrlPIDSys* are only default values, which will be overwritten because they are parameters under study.

2.3 Creating Basic Models with OpenModelica

OpenModelica defines a set of basic models for different fields of engineering. An OpenModelica *package* is created that contains all basic models. Although this leads to a duplication of some components, this makes the model base independent of potential future changes in OpenModelica libraries. The package is filled with the following basic models whose names correspond to the names in the mb-attributes of the leaf nodes in the SES:

- *Constant* as the setpoint for the controlled variable
- *Step* for stimulating the disturbance
- *Feedback* for closing the feedback control loop
- *PID* is the controller of the feedback control system
- *TransferFunction* for representing the process, the disturbance's behavior, and the feedforward
- *Add* for adding signals

Each basic model can be configured according to the attributes of the leaf node which they are linked to in the SES. The package acts as an MB for OpenModelica basic models.

2.4 Experiment Execution

For executing simulation based experiments the experiment process and its goals need to be defined in a Python script. This script implements the EC according to Figure 2. The Python framework provides some EC related template scripts. The goals of the experiment were discussed in Section 2.1. The experiment should start with the study of different PID controller configurations using the control system structure without feedforward controller. In case that the objectives are not achieved by just varying the parameters k and Ti of the PID controller, the study shall be carried out with the additional feedforward control structure. A snippet of the EC script with essential steps of the experiment process is given next.

```
...
SESfile = ...
if conditions_for_experiment:
  #prune, flatten, and build
  SESvar = {"mysim": "OpenModelica",
       "feedforward": 0}
  PESfile = SESToPy("prune",SESvar,
       SESfile)
  FPESfile = SESToPy("flatten",PESfile)
  smHandle = SESMoPy("build",FPESfile)
  #execute
  sim_param = ...
  results = SESEuPy("simulate",smHandle)
...
elif conditions_for_experiment:
  #prune, flatten, and build
  SESvar = {"mysim": "OpenModelica",
       "feedforward": 1}
  PESfile = SESToPy("prune",SESvar,
       SESfile)
  FPESfile = SESToPy("flatten",PESfile)
  smHandle = SESMoPy("build",FPESfile)
  #execute
  sim_param = ...
  results = SESEuPy("simulate",smHandle)
...
```

The EC starts the experiment by setting the SESvar *mysim* and $feedforward$. Next, the EC calls SESToPy's API method for pruning with the current SESvar values and a reference to the file defining the SES as JSON structure. The pruning process results in a PES coded as JSON structure. Afterwards, the EC calls SESToPy's API method for flattening the PES. The created FPES is similar to the FPES shown in Figure 5, which represents the more complex FPES for the later SESvar assignment $feedforward = 1$. A reference to the file containing the FPES as a JSON structure is returned to the EC. The EC then calls SESMoPy's API method for the build method and passes the FPES file handle. SESMoPy determines the target simulator from the attribute at the node *simMethod* and the value ranges of the PID controller parameters under study from the attribute at node *expMethod* in the FPES.

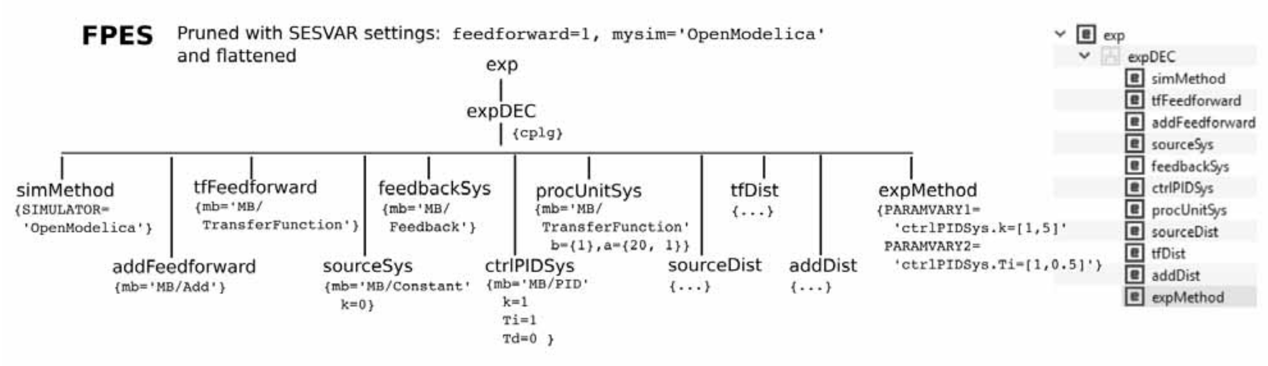

Figure 5: Left: FPES to study the feedback control system structure with feedforward; Right: FPES representation in SESToPy.

Based on the information in the FPES and the basic models from the MB, SESMoPy creates multiple files. For each configuration of the simulation model of the control system one target simulator specific file containing the model and its configuration is created. Simulation models of one structure variant have different configurations of the PID controller. Furthermore a configuration file containing information about the created models and the target simulator is created. A handle to all the created files is returned by SESMoPy to the EC, referred to as *smHandle*. The EC extends the configuration file with simulation data, such as the solver to use, etc. Then, the EC calls the tool SESEuPy and passes the smHandle as the link to the model files and the configuration file. In collaboration with the target simulation environment OpenModelica, SESEuPy creates the fully configured simulation model and controls its execution. Figure 6 shows the structure of a fully configured OpenModelica model, but with feedforward controller, i.e. for the SESvar assignment $feedforward = 1$. Finally, SESEuPy returns the simulation results to the EC.

In case the results meet the experimental goals, the overall results are calculated and returned by the EC. Otherwise a new model configuration and generation needs to be started. For this purpose, the $elif$ part in the code snippet defining the EC defines the experiment steps for the second system structure with the additional feedforward controller by the SESvar assignment $feedforward = 1$.

If the experimental goals have been achieved, the overall results of the experiment are the necessary control structure and the appropriate PID controller parameter settings. Otherwise the failure to achieve the objectives may also be established.

Figure 7 depicts the simulation results of the feedback control system resulting from the execution of the presented EC. In parts (a) and (b) the simulation results of the system structure without feedforward control and with different PID controller parameter settings are presented. The required control goals of a settling time of less than 15 seconds and a maximum overshoot of less than 5% after a disturbance as specified in Section 2.1 for the tested PID controller parameter settings could not be achieved. In part (c) of Figure 7 the simulation results of the system structure with feedforward control is presented. The required control goals are reached and the PID controller parameter settings are returned as overall results.

3 Conclusion and Future Work

In this paper free software tools for variant modeling are presented, beginning with the system specification with an SES up to automatic variant derivation, model building and execution. The eSES/MB infrastructure has been implemented as prototype Python tools. They make it possible to model and simulate engineering problems using different target simulation environments. Future works will especially cover support for physical modeling, a deeper integration of FMI into SESMoPy and test other target simulation environments.

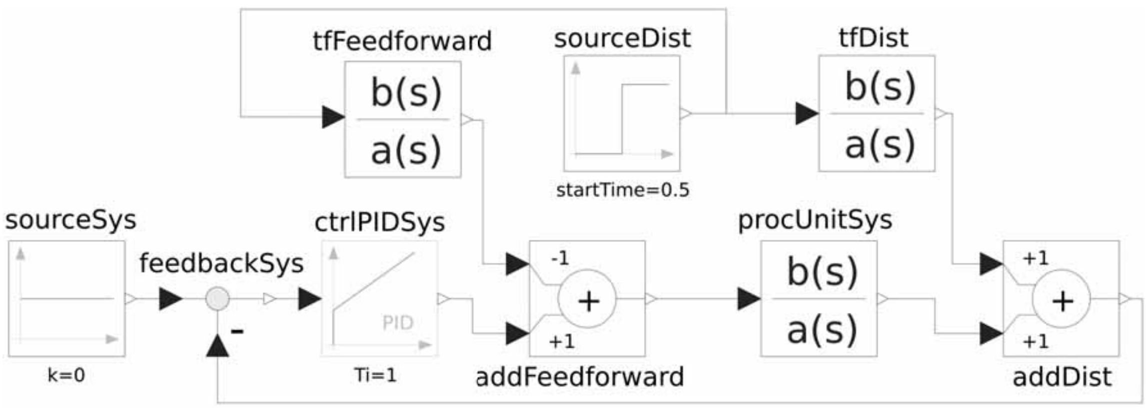

Figure 6: OpenModelica SM of the feedback control system with feedforward control.

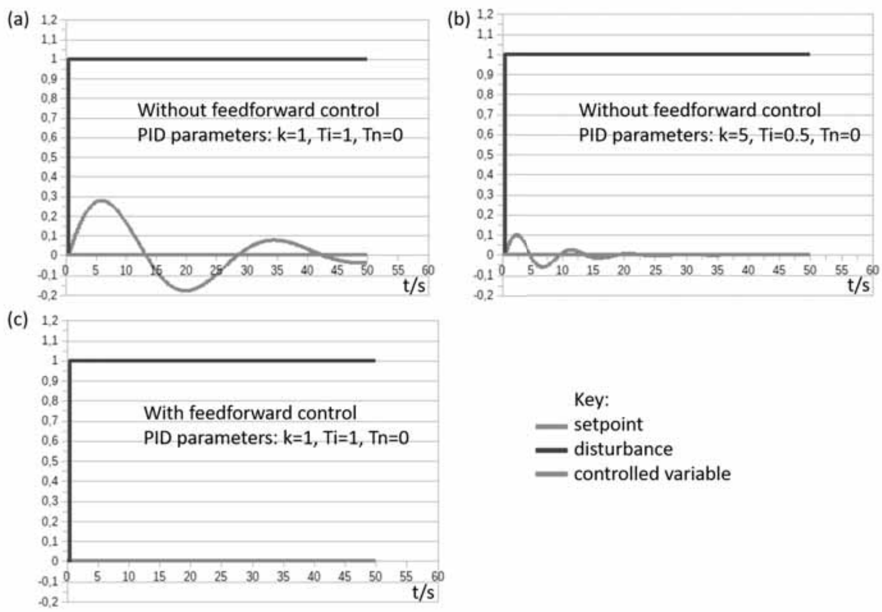

Figure 7: (a) and (b): Without feedforward control and with different PID controller settings, (c): With feedforward control.

Acknowledgement

The authors acknowledge the grant from the German Science Foundation, DFG (PA 631/2). Moreover, the authors would like to thank Peter Junglas, who contributed valuable work to the development of a generic model builder, Daniel Pascheka, who implemented a first version of the graphical editor for MATLAB, Birger Freymann, who redesigned the MATLAB editor and implemented a model builder for the MatlabDEVS toolbox, and our former colleague, Tobias Schwatinski, who provided preliminary work. Last, but not least, we would like to thank Bernie Zeigler for motivating and supporting our work.

References

[1] Folkerts H, Deatcu C, Pawletta T, Hartmann S. A Python Framework for Model Specification and Automatic Model Generation for Multiple Simulators. *2019 International Interdisciplinary PhD Workshop*; 2019 May; Wismar. IEEE. doi: 10.1109/IIPHDW.2019.8755423.

[2] Kang KC, Cohen SG, Hess JA, Novak WE, Peterson AS. *Feature-Oriented Domain Analysis (FODA) Feasibility Study.* Technical Report CMU/SEI-90-TR-21, ESD-90-TR-222, SE Inst. Carnegie Mellon Univ. Pittsburgh/PA, USA. 1990.161 p.

[3] pure-systems GmbH. *Technical White Paper: Variant Management with pure::variants.* 2006. https://www.pure-systems.com/mediapool/pv-whitepaper-en-04.pdf, last accessed 2019/08/06.

[4] Mellor SJ, Balcer M. *Executable UML: A Foundation for Model-Driven Architectures.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.; 2002.

[5] Zeigler BP, Kim TG, Praehofer H. *Theory of Modeling and Simulation.* 2nd ed. Cambridge: Academic Pr.; 2000. 510 p.

[6] Pawletta T, Schmidt A, Zeigler BP, Durak U. Extended Variability Modeling Using System Entity Structure Ontology Within MATLAB/Simulink. In Proc. of Spring Simulation Multi-Conference 2016. *Spring Simulation Multi-Conference*; 2016 Apr; Pasadena/CA, USA. SCS Int. p 62-69.

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

[8] Pawletta T, Pascheka D, Schmidt A, Pawletta S.Ontology-Assisted System Modeling and Simulation within MATLAB/Simulink. *SNE Simulation Notes Europe.* 2014; 24: 59–68. doi: 10.11128/sne.24.tn.102241.

[9] Research Group CEA. *Python-Based eSES/MB Infrastruture.* 2019. http://www.cea-wismar.de/tbx/Py_eSESMB/, last accessed 2019/08/06.

[10] Capocchi L, Santucci JF, Poggi B, Nicolai C. DEVSimPy: A Collaborative Python Software for Modeling and Simulation of DEVS Systems. *20th International WETICE Conference.* 2011 June; Paris. 170–175. doi: 10.1109/WETICE.2011.31.

[11] Modelica Association Project "FMI". *Functional mock-up interface for model exchange and co-simulation.* 2019. https://fmi-standard.org/tools/, last accessed 2019/08/06.

[12] Modelica Association Project "FMI". *Functional Mock-up Interface for Model Exchange and Co-Simulation.* 2014. https://svn.modelica.org/fmi/branches/public/specifications/v2.0/FMI_for_ModelExchange_and_CoSimulation_v2.0.pdf, last accessed 2019/08/06.

[13] Schmidt, A. *Variant Management in Modeling and Simulation Using the SES/MB Framework* [dissertation]. University of Rostock; 2019.

[14] Deatcu C, Folkerts H, Pawletta T, Durak U. Design Patterns for Variability Modeling Using SES Ontology. In Proc. of Spring Simulation Multi-Conference 2018. *Spring Simulation Multi-Conference*; 2018 Apr; Baltimore/MD, USA. SCS Int. p 3:1–3:12.

SNE Simulation News

EUROSIM Data and Quick Info

Contents

Short Info EUROSIM N2
Short Info ASIM, CEA-SMSG N3
Short Info CSSS, DBSS, LIOPHANT, LSS N4
Short Info KA-SIM, MIMOS, NSSM, PSCS N5
Short Info SIMS, SLOSIM, UKSIM N6
Short Info ROMSIM, Albanian Society N7
Short Info ARGESIM, SNE N8
Snapshots EUROSIM Congress 2019 .... Cover Inside

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, and president, past president, and **SNE** representative. 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	Niki 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 peer reviewed contributions as well as a membership journal for **EUROSIM** with information from the societies. **EUROSIM** societies distribute **SNE** to their members 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 and Conferences.

Each year a major **EUROSIM** event takes place:

EUROSIM Congress. Every three years the **EUROSIM 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 2022, the *11th EUROSIM Congress*, will be organised by **DBSS**, the Dutch Benelux simulation society, in Amsterdam, July 2022. → *www.dutchbss.org*

MATHMOD Vienna. This triennial conference is mainly organized by **ASIM**, the German simulation society, with main co-sponsors **EUROSIM** and **IFAC**.
MATHMOD 2021, *the 10th MATHMOD Vienna Conference on Mathematical Modelling* will take place in Vienna, Februray 24-26, 2021. → *www.mathmod.at*

SIMS EUROSIM Conference. SIMS, the Scandinavian simulation society starts in 2020 with this conference series, organising every third year the annual **SIMS Conference** on a broader basis as **SIMS EUROSIM Conference**. Next is *SIMS 2020 EUROSIM Conference on Modelling and Simulation*, September 22-25, 2020, Oulu, Finland.
→ *www.scansims.org*

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, *vadime@wolfram.com*
Repr. EUROSIM	Erik Dahlquist, *erik.dahlquist@mdh.se*
Edit. Board SNE	Esko Juuso, *esko.juuso@oulu.fi*
Web EUROSIM	Vadim Engelson, *vadime@wolfram.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
18.-20. 9. 2019; M. Putz, A. Schlegel (Hrsg.), Verlag Wissenschaftliche Skripten Auerbach, 2019, ISBN print 978-3-95735-113-5, ISBN ebook 978-3-95735-114-2; ASIM Mitteilung 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

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 (Hrsg.); ISBN 978-3-642-14535-3, Springer, 2011; AM 130

Book Series Fortschrittsberichte Simulation – Advances in Simulation

Artur Schmidt: **Variantenmanagement in der Modellbildung und Simulation unter Verwendung des SES/MB Frameworks**. FBS 30; ISBN ebook 978-3-903347-30-4, DOI 10.11128/fbs.30, ARGESIM Wien, 2019; ISBN print 978-3-903311-03-9, TUVerlag Wien (print on demand), 2019

Martin Bicher: **Classification of Microscopic Models with Respect to Aggregated System Behaviour**. FBS 29; ISBN ebook 978-3-903347-29-8, DOI 10.11128/fbs.29, ARGESIM Wien, 2017; ISBN print 978-3-903311-00-8, TUVerlag Wien (print on demand), 2019

Martin Bachler: **Model Based Methods for Early Diagnosis of Cardiovascular Diseases**. FBS 28; ISBN ebook 978-3-903347-28-1, DOI 10.11128/fbs.28, ARGESIM Wien, 2017 ISBN print 978-3-903024-99-1, TUVerlag Wien (print on demand), 2019

Andreas Körner: **A Mathematical Characterisation of State Events in Hybrid Modelling.** FBS 27; ISBN ebook 978-3-903347-27-4, DOI 10.11128/fbs.27, ARGESIM Vienna, 2016

Nikolas Popper: **Comparative Modelling and Simulation: A Concept for Modular Modelling and Hybrid Simulation of Complex Systems.** FBS 26; ISBN ebook 978-3-903347-26-7, DOI 10.11128/fbs.26, ARGESIM 2016

Gunnar Maletzki: **Rapid Control Prototyping komplexer und flexibler Robotersteuerungen auf Basis des SBE-Ansatzes**. FBS 25; ISBN ebook 978-3-903347-25-0, DOI 10.11128/fbs.25, ARGESIM Wien, 2019; 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 ebook 978-3-903347-24-3, DOI 10.11128/fbs.24, ARGESIM Vienna, 2016

Martin Bruckner: **Agentenbasierte Simulation von Personenströmen mit unterschiedlichen Charakteristiken.** FBS 23; ISBN ebook Online 978-3-903347-23-6, DOI 10.11128/fbs.23, ARGESIM Wien, 2016

Stefan Emrich: **Deployment of Mathematical Simulation Models for Space Management.** FBS 22; ISBN ebook 978-3-903347-22-9, DOI 10.11128/fbs.22, ARGESIM Publisher Vienna, 2016

Xenia Descovich: **Lattice Boltzmann Modeling and Simulation of Incompressible Flows in Distensible Tubes for Applications in Hemodynamics.** FBS 21; ISBN ebook 978-3-903347-21-2, DOI 10.11128/fbs.21, ARGESIM Vienna, 2016; ISBN Print 978-3-903024-98-4, TUVerlag Wien (print on demand), 2019

Florian Miksch: **Mathematical Modeling for New Insights into Epidemics by Herd Immunity and Serotype Shift.** FBS 20; ISBN ebook 978-3-903347-20-5, DOI 10.11128/fbs.20, ARGESIM Publisher Vienna, 2016; ISBN Print 978-3-903024-21-2, TUVerlag Wien (print on demand), 2016

Shabnam Tauböck: **Integration of Agent Based Modelling in DEVS for Utilisation Analysis: The MoreSpace Project at TU Vienna.** FBS 19; ISBN ebook 978-3-903347-19-9, DOI 10.11128/fbs.19, ARGESIM Vienna, 2016; ISBN Print 978-3-903024-85-4, TUVerlag Wien (print on demand), 2019

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

www.tuverlag.at
ISBN 978-3-903024-97-7