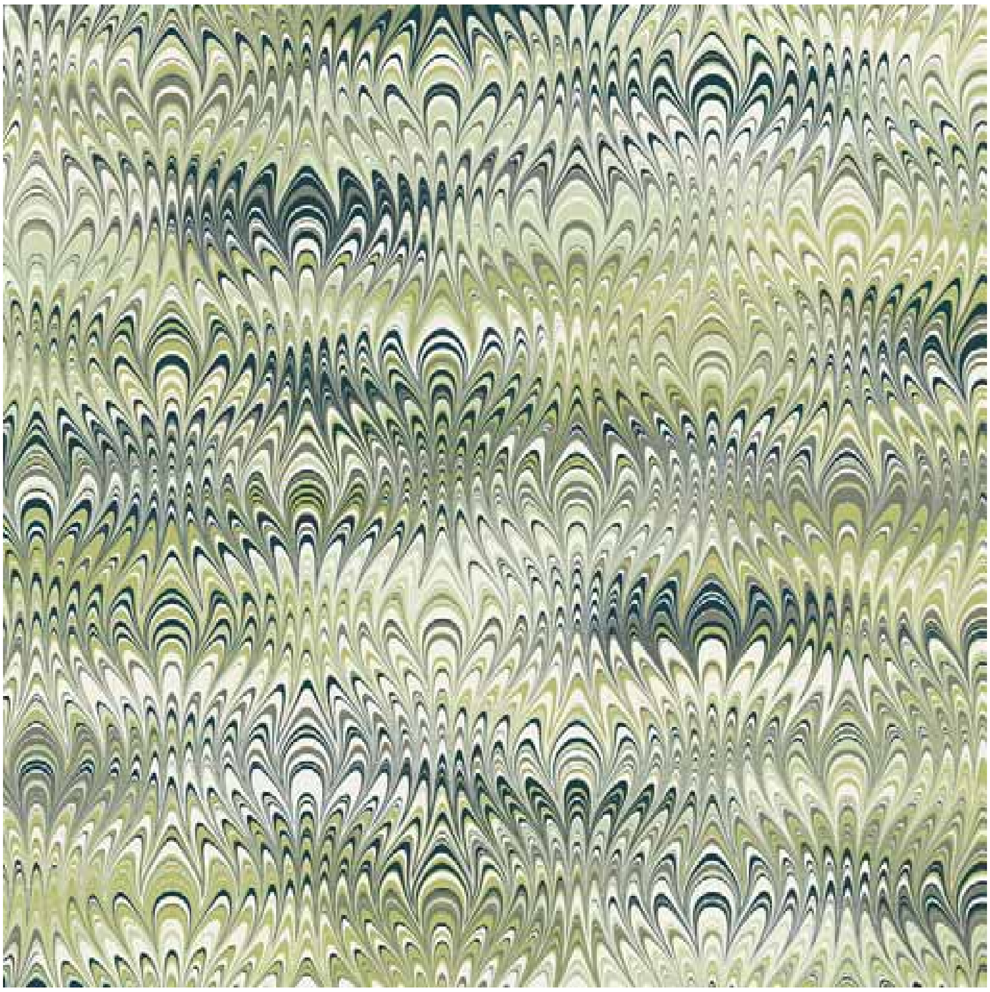
SNE SIMULATION NOTES EUROPE



Journal on Developments and Trends in Modelling and Simulation

EUROSIM Scientific Membership Journal

Vol. 34 No.1, March 2024
ISSN Online 2306-0271
ISSN Print 2305-9974
DOI 10.11128/sne.34.1.1067

ARGESIM

ASIM 2024

27th Symposium Simulation Technique

September 4-6, 2024, Universität der Bundeswehr, Munich, Germany

www.asim-gi.org/asim2024

2nd SIMS EUROSIM 2024 and 65th SIMS 2024

September 10-12, 2024, Oulu, Finland

www.scansims.org

I3M 2024

Int. Multidisciplinary Modeling & Simulation Multiconference

September 18-20, 2024, Tenerife, Spain

www.msc-les.org/i3m2024

2024 Winter Simulation Conference

December 15-18, 2024, Orlando, Florida

www.wintersim.org

MATHMOD 2025

11th Vienna Int. Conference on Mathematical Modelling

February 19-21, 2025, Vienna, Austria

www.mathmod.at

EUROSIM Congress 2026

July 2026, Italy

www.eurosim.info

Editorial

Dear Readers, *This first issue of SNE Volume 34, 2024, SNE 34(1), shows the variety of SNE contribution types: Technical Notes (with submitted contributions and post-conference contributions), Software Notes (contributions on development of software for modelling and simulation), and Benchmark Notes (contributions with case studies for an ARGESIM Benchmark).*
First, D. Lückerath and O. Ullrich's contribution (a submitted Technical Note) present a simulation model for increasing the resilience in public transit networks. Then, A. Naja et al. present the development of a scenario toolkit for autonomous systems (a Technical Note as post-conference publication from an ASIM event). Next, A. Herzog provides two Benchmark Notes with case studies for Benchmark 'C2 –Flexible Assembly System' and 'C22 – Non-standard Queuing Policies' with the tool 'Warteschlangensimulator'.
Then M. Hempel and J. Heger present a simulation-based investigation of energy flexibility in the optimization of hinterland drainage (a Technical Note as English post-conference publication from an ASIM event). And finally, A. Wuttke et al. introduce LogFarm, an open source simulator for logistics networks (a Software Note, including also a benchmark study for Benchmark 'C14 – Supply Chain'). This contribution is based on A. Wuttke's master thesis, which won the simulation price at the ASIM Conference on Simulation in Production and Logistics 2023 in llmenau. Further contribution types as Overview Note, Short Notes, Education Notes and Project Notes (more info on next page) will be part of the next SNE issues.
The cover of this SNE continues with digital marbling graphics by Graham Horton – algorithm art modelling and simulating the handcraft of making marbled paper (used for covers in traditional bookbinding) with old and new patterns.
I would like to thank all authors for their contributions, and many thanks to the SNE Editorial Office for layout, typesetting, preparations for printing, electronic publishing, and much more. And have a look at the info on EUROSIM-related simulation events this year: ASIM Symposium in Munich, I3M conference in Tenerife, EUROSIM-SIMS conference in Oulu, and WinterSim in Orlando.

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

Contents SNE 34(1)

e-SNE 34(1), DOI 10.11128/sne.34.1.1067

www.sne-journal.org/sne-volumes/volume-34

SNE Basic e-Version with Open Access

SNE Full e-Version for Members of publication-active EUROSIM Societies: ASIM, CEA-SMSG, CSSS, DBSS, KA-SIM, LIOPHANT, NSSM, PTSK, SIMS, SLOSIM, UKSIM

print-SNE for members on demand (printer INTU TU Wien)

Combining Pre-Event Planning and Emergency Response in a Simulation Model to Increase the Resilience in Public Transit Networks. *D. Lückerath, O. Ullrich* 1

Towards a Scenario Toolkit for Autonomous Systems *A. Naja, S. Gupta, U. Durak, S. Hartmann* 13

A Discrete Process Modelling Study of ARGESIM Benchmark 'C2 -- Flexible Assembly System' with Warteschlangensimulator. *A. Herzog* 23

A Discrete Process Modelling Study of ARGESIM Comparison 'C22 -- Non-standard Queuing Policies' with Warteschlangensimulator. *A. Herzog* 29

Simulation-based Investigation of Energy Flexibility in the Optimization of Hinterland Drainage. *M. Hempel, J. Heger* 35

LogFarm: An Open Source Graph-based Simulator for Logistics Networks. *A. Wuttke, J. Hunker, M. Rabe* 43

EUROSIM Societies & ARGESIM/SNE Short Info . N1 – N4

Conferences EUROSIM / ASIM .... Covers

Cover: Digital Marbling by Graham Horton, type 'Ikarus'; *digital-marbling.de*

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
Felix Breitenecker (Organisation, Author Mentoring)
Irmgard Husinsky (Web, Electronic Publishing),
Johannes Tanzler (Layout, Organisation),
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: INTU (TU Wien), Wiedner Hauptstrasse 8-10, 1040, Vienna, Austria – *www.intu.at*

ARGESIM Publisher: 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 (www.sne-journal.org).

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. **SNE** follows the recent developments and trends of modelling and simulation in new and/or joining 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.

SNE is the scientific membership journal of **EUROSIM**, the *Federation of European Simulation Societies and Simulation Groups* (www.eurosim.info), also providing Postconference publication for events of the member societies. **SNE**, primarily an electronic journal **e-SNE** (ISSN 2306-0271), follows an open access strategy, with free download in basic version (B/W, low resolution graphics). Members of most **EUROSIM** societies are entitled to download **e-SNE** in an elaborate full version (colour, high resolution graphics), and to access additional sources of benchmark publications, model sources, etc. (via group login of the society), **print-SNE** (ISSN 2305-9974) is available for specific groups of **EUROSIM** societies.

SNE is published by **ARGESIM** (www.argesim.org) on mandate of **EUROSIM** and **ASIM** (www.asim-gi.org), the German simulation society. **SNE** is DOI indexed with prefix 10.11128.

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.
- **SN** Short Note, max. 5 p.
- **SW** Software Note , 4–6 p.
- **BN** Benchmark Note, 2–10 p.
- **ON** Overview Note – only upon invitation, up to 14 p.
- **EN** Education Note, 6–8 p.
- **PN** Project Note 6–8 p.
- **STN** Student Note, 4-6 p., on supervisor's recommendation
- **EBN** Educational Benchmark Note, 4–10 p.

Further info and templates (doc, tex) at **SNE**'s website, or from the Editor-in-Chief

www.sne-journal.org

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

SNE Editorial Board

SNE - Simulation Notes Europe is advised and supervised by an international scientific editorial board. This board is taking care on peer reviewing of submission to SNE (and extended for special issues and Postconference publication):

Combining Pre-Event Planning and Emergency Response in a Simulation Model to Increase the Resilience in Public Transit Networks

Daniel Lückerath, Oliver Ullrich

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

SNE 34(1), 2024, 1-12, DOI: 10.11128/sne.34.tn.10671
Received: 2023-12-5; Revised: 2024-02-07
Accepted: 2024-02-15
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. With increasing pressure from the impact of climate change, urban transit operators aim to improve their networks' resilience against both small disturbances and larger disruptions and outages. To be applicable in pre-planning or emergency response, such measures and strategies have to be thoroughly evaluated. To aid this evaluation process, this paper presents different operational strategies – and their systematic evaluation using a bi-modal transit simulation model – designed to increase the resilience of urban transit networks against the impact of climate change. To illustrate the application of such a system, the paper examines the strategies, including both delay management and disruption management measures based on a simulation model of the urban transit network of the city of Cologne, Germany.

Introduction

Urban transit infrastructure is increasingly put under pressure by the impact of climate change [1]. In general, there are two types of strains: Continuous, slowly increasing stress, which creates more and more small disturbances, and extreme events that lead to larger outages. Out of the latter category, most relevant for European urban centers are coastal, fluvial, and pluvial flooding, flash floods caused by heavy precipitation, rockslides and landslides, temperature extremes, thunderstorms and tornados, winter storms, and rising sea levels [2]. For the former category, the specific impact and its cost are complex to measure exactly.

However, it is being estimated that on average 30% to 50% of current road maintenance cost is already seen as consequences of climate change [3].

To be resilient against both types of stress infrastructure, including urban transit networks, needs a combination of disaster risk management and resilience-increasing strategies, i.e., a combination of good planning in the strategic timeframe and good emergency management in the operational timeframe. To be effective and efficient, both short-term and long-term activities have to be able to rely on well-evaluated operational strategies, i.e., well-tested combinations of measures to mitigate and adapt to the impact of climate change.

Here, simulation can help: In case of sudden disasters impacting transit systems, including extreme weather and human-made events, operators have to be able to make decisions fast to a) transfer the infrastructure components into a pre-planned disaster mode and b) to be able to re-establish services as soon as the immediate event has passed.

These operators can be assisted with a simulation application covering both light rail and bus transit that executes simulation runs sufficiently fast to enable evaluation and comparison of potential decisions and operational strageties, thereby contributing to increase the resilience of the transit system. The same simulation model can be used for 'what-if' analysis in the context of strategic planning of adaptation strategies against the slowly increasing impact of climate change.

This paper presents the application of a bi-modal transit simulation framework to evaluate different operational strategies – both in pre-planning and in emergency response – designed to increase the resilience of urban transit networks against the impact of climate change in general, and of extreme weather events more specifically.

Instead of looking at purely artificial use cases, the paper examines the strategies and their combinations based on a simulation model of the urban transit network of the city of Cologne, Germany.

The paper continues with sharing some background on different concepts of resilience in a public transit concept and on operational strategies aimed at increasing transit resilience (Section 1). It then describes public transit modeling in general and the applied transit simulation model (Section 2). The main part begins with a short description of Cologne's public transit network and continues with the test and evaluation of several operational strategies aimed at combining pre-event planning and emergency response in the context of that transit network (Section 3 and Section 4). The paper concludes with a short discussion of the lessons learned and an outlook to further research (Section 5).

1 Resilience in Public Transit

1.1 Urban Transit Resilience

In the urban transit context, two different understandings of 'resilience' are relevant: engineering (or 'narrow') resilience and multi-equilibria resilience [4][5][6]. Engineering resilience aims at stability and control, i.e., to withstand shocks and to return to the stable pre-disaster state as fast as possible ('bouncing back', see e.g., [7]). Subsequently, the concept of engineering resilience is static and does not take the need for flexibility and adaptation into account. Multi-equilibria resilience [5] on the other hand acknowledges that a disturbed system might not always return to the same stable pre-disaster state and aims at adapting the system to better cope with the disaster ('bouncing forward').

For urban transit systems to withstand different types of disasters, transit operators need to design schedules and networks with both resilience concepts in mind. While engineering resilience is useful for mitigating small to medium disturbances that inevitably happen during an operational day (e.g., passengers holding open doors for other passengers), multi-equilibira resilience becomes relevant when addressing medium to large disturbances that might require extensive (temporary) modifications of schedules and vehicle routes.

Engineering resilience is usually addressed as part of the medium- to long-term planning (e.g., by designing schedules with high regularity of departure times [8]), multi-equilibria resilience can additionally be addressed in the short- to medium-term planning (e.g., by rerouting vehicles or purposely delaying departure times to keep transfer connections between different transit modes).

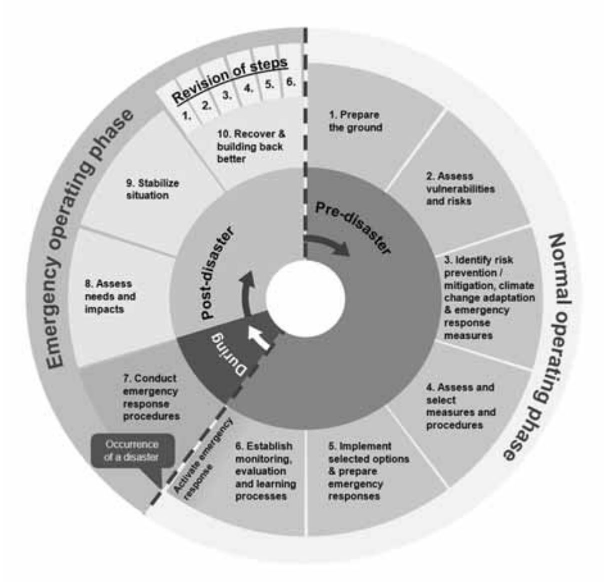


Figure 1: The combined Disaster Risk Management and Resilience Improvement Cycle (Source: [11][12]).

Considering accelerating climate change, the associated increase in frequency and intensity of natural disasters, and the subsequent increase in impacts to (urban transit) infrastructure [9][10], it becomes paramount to design new schedules and networks in a resilient and sustainable way, and to address both types of resilience. From a procedural point of view transit providers can accomplish resilience improvement by adopting a combined disaster risk management and climate change adaptation cycle [11], encompassing both long-term planning of services during normal operations and short-term disaster management during emergencies (see Figure 1).

1.2 Operational Measures to Increase Resilience

Operators can apply a variety of strategies to increase both engineering and multi-equilibria resilience. These measures generally fall into one of two categories: *Delay management measures* are designed to increase service regularity and vehicle punctuality by applying timetable- and rule-based holding strategies. *Disruption management measures* apply more comprehensive interventions that change line routes and schedules of multiple vehicles to mitigate the impact of larger disruptions of the network.

The following paragraphs give brief introductions on delay management and disruption management measures, with a more detailed discussion found in [13].

Delay management strategies. In day-to-day operations, significant effort is made to ensure service quality by avoiding vehicle bunching. Vehicle bunching describes the phenomenon of public transit vehicles to form pairs due to a preceding delayed vehicle taking on more passengers as originally planned and the on-time succeeding vehicle subsequently taking on fewer passengers as originally planned. Thus, without intervention the preceding vehicle gets further delayed while the succeeding vehicle catches up to it [14]. Basic *bunching mitigation measures* hold back a vehicle at a time control point if it is too early by a certain threshold. A more complex version of that strategy also considers a vehicle's estimated punctuality at its next scheduled stops down the line [15]).

The other major category of delay management strategies are *synchronization assurance measures*, aimed at ensuring transfers from one line to another. Usually, these transfers are either rendezvous connections, where a number of vehicles serving different lines wait at the same station to enable passengers to transfer to each other, or directed transfer connections, where a vehicle serving a line waits for a feeder vehicle serving a different line, and thus enables transfer from the second line to the first. To assure these transfer connections, one or more vehicles might have to be kept at a station to wait for a delayed vehicle.

Disruption management strategies. Disruption management measures are usually much more comprehensive and more incisive than delay management strategies. They cover rerouting, short-turning, stop skipping, and route separation. Typically, the measures are ordered by their degree of intervention to form an overall disruption management strategy.

If a disruption occurs that cannot be migitated by delay management, first rerouting of vehicles is considered: sets of potential alternative routes for each line affected by the disruption are constructed using a pathfinding algorithm. Once all potential alternative routes are determined, the method picks – depending on the operator's preferences – either the route with the least traversal time or the most punctual route.

If no alternative route covering all regular stops can be found, short-turning, i.e. ending the trip before reaching the disrupted network section, is considered. Such an action has an impact on the executing vehicle's next trip that has to be mitigated as well:

- Either the vehicle has to make a deadheading trip to the first stop of its next trip,
- or the next trip of the vehicle has to be short-turned as well to start from the vehicle's current position.

A potential disadvantage of the described short-turning method is that it does not guarantee the reachability of all stops further down the route. Sometimes that can be remedied by skipping a part of the stops on the original route.

If all these measures fail, an operator will, where possible, separate all affected routes in the disrupted network section. This can be viewed as a two-sided short-turning strategy, where vehicles stuck on either side of the disrupted network section service as much of their originally scheduled trips as possible. In addition to just short-turning affected trips, route separation generally also requires adjustments to the timetable and the vehicle schedule.

2 Modeling Urban Transit

2.1 Urban Public Transit Modeling

Urban transit consists of a number of interacting networks, e.g., a light rail system, express and community bus networks. Such a network is based on street and rail segments as well as stops and stations where passenger exchanges take place. These stops and stations are served by a set of transit vehicles executing service trips by following pre-defined routes through the network. During the operational day each individual vehicle executes a sequence of service trips, interspersed with deadheads, that is called a rotation. The rotation schedule defines the assignment of specific vehicles to rotations.

While some stops, mainly bus stops, include a bay with capacity for more than one vehicle, many other stops can contain only one vehicle at any given time. Some stops are marked as control points, i.e., locations in the network where control strategies may be employed, e.g., purposely delaying early vehicles until the scheduled departure time is reached. At other stops, vehicles depart as soon as the passenger exchange is completed. Directed paths through the network, connecting two successive stops are called connections.

They usually consist of several street and/or rail segments, junctions, and signals, that in turn can be shared by several connections. Access to individual segments is controlled by signals, usually at junctions. Often, two or more signals constitute a signal group with a common scheduling strategy.

Typically, daily operations are managed by an operations center, with dispatcher personnel managing procedures for the mitigation of small disturbances and larger outages. In case of any disturbances or outages, transit operators have several remedies at their disposal to keep services running as long as possible, and to restore them as soon as possible. These include the authority to short-turn or cancel trips, to re-route vehicles, and to deploy extra vehicles.

Simulation models that represent the described entities and behavior are often extensions of already established models of individual traffic [16][17][18]. Generally, many of the more recent simulation models including bus transit use microscopic agent-based modeling approaches [16][17][19][20][21], the mesoscopic approach to bus transit simulation proposed by Toledo et al. [18] extends a mesoscopic simulation model for individual traffic based on queuing theory proposed by Burghout [21], which represents the street network as a graph of interconnected queues and vehicles as individual entities traversing these queues based on speed/density functions.

Especially models utilizing a fine-grained modeling approach generally necessitate the availability of an extensive data basis, including detailed information on origin-destination matrices, vehicular dynamics, signaling strategies, and lane changing rules [22], and include many components which are not immediately interesting for public transit resilience management. This often leads to long runtimes [23][24], thereby rendering those models inadequate for short-term disaster management. Therefore, this paper applies the fast mesoscopic transit simulation model described by Ullrich and Lückerath [25] and [26].

2.2 A Mesoscopic Model of Multi-Modal Public Transit

To examine cost and benefits of different resilience-increasing strategies a mesoscopic urban transit simulation model has been developed based on the event-based approach [27]. Described in detail by Ullrich and Lückerath [25] we now only give a short overview of its characteristics.

At the center of the model lies the representation of the physical transit network as a directed graph. Stops, connections and segments are modeled as nodes of this graph, with their neighborhood relations modeled as edges. Each node has a geographic position, identifying attributes, and a maximum vehicle capacity.

To represent the driving behavior of different traffic modes, the model distinguishes between two types of segment nodes: roads and tracks. Road nodes are segment nodes that are used by entities of individual traffic, have an unrestricted vehicle capacity, and do not enforce a fixed vehicle sequence. Track nodes are used exclusively by rail vehicle entities and enforce both compliance with a maximum vehicle capacity as well as a fixed vehicle sequence.

Each node represents an entity in the sense of the event-based simulation paradigm, i.e., it can be producer and consumer of events. Thus, temporary changes of attribute characteristics, e.g., for modeling disruptions, can be mapped in a simple way via events and activities.

Vehicles are represented as transient entities that encapsulate a significant portion of the event-based simulation logic and move across the model graph during a simulation run. Each vehicle entity has a reference to the trip it is currently serving, i.e., at each simulation time it only has access to the information that is directly relevant for its current activity. In the model, vehicles are classified according to their transit mode, their vehicle type, and their individual vehicle characteristics.

In addition to the physical network components and vehicle behavior presented so far, concepts such as lines, trips and timetables also are represented in the model.

To allow for management on a higher level than individual trips, the timetable must be supplemented by a rotation schedule, which combines trips into groups (so called rotations) [28] that can be executed by individual vehicles within an operating day. These and other management activities are encapsulated in three management modules: the fleet manager, the line manager, and the dispatcher. Thus, changes to the modeling of individual administrative activities do not affect the modeling of other areas of the simulation model. Work in progress on these modules has been reported in [29] and [26].

2.3 Generating Regular Timetables Adhering to Planning Requirements

Transit timetable generation is a well-researched complex optimization problem [30], too complex to describe here in detail. Generally, to accomplish resilience against small disturbances optimization models often aim for *service regularity* [31][32][33][8], a measure of the equability of headways that can be used for static evaluation of a timetable during the planning phase as well as for dynamic assessment of operational performance.

In addition to being resilience against small disturbances, a feasible schedule also has to adhere to other planning requirements – that includes specific departure sequences to accommodate frequent transfer connections or deliberately short headways to reduce the passenger load of follow-up vehicles.

This study applies a disjunctive program formulation producing regular timetables for multi-modal public transit systems adhering to planning requirements given by transit operators. That program was first introduced by Lückerath, Ullrich, Rishe, and Speckenmeyer [33] and allows for the consideration of feasibility constraints from daily operations as well as for the consideration of simultaneous departures for transfer connections, an objective traditionally opposed to regularity.

3 Cologne's Bi-Modal Urban Transit Network

The city of Cologne's urban transit service is organized as a combined bus and light-rail network. Generally, the light-rail lines transport residents and commuters inside the densely populated inner city, as well as connects the inner city to suburban outskirts and neighboring towns. Both functions are highly relevant for daily commuters.

Parts of the light-rail network run overground, in some parts light rail cars share the roadway with car traffic, in others they have exclusive rights of way as well as signal precedence compared to individual traffic. Other light-rail segments run underground. In contrast to many other European cities, above- and below-ground railroads are not separated; instead, the Cologne subway behaves more like an underground streetcar than a typical subway – it does not use specific underground engines or passenger cars with their wide aisles and comparatively few seats for the typically short subway journeys, no turnstiles exist at the platforms.

The bus network includes both express buses on their own right of way and relatively slow community buses that connect neighboring districts as well as provide intra-district connections.

Bus and light-rail network have in common strategic nodes that allow for transfer between the networks, including the stations Barbarossaplatz, Ebertplatz, and Neumarkt. These nodes are usually time control points.

Additionally, at some nodes the urban transit network is connected to national rail, that includes the stations Deutz Bf, Ehrenfeld Bf, Hansaring Bf, Hauptbahnhof, and Mülheim Bf.

In total, the Cologne urban transit network consists of 1,770 stops, of which 1,242 are serviced by buses and 528 by light-rail trains of the types Vossloh Kiepe K4000 [34], K4500 [35], and K5000 [36]. The vehicles service 68 bus and 30 light-rail routes. The light rail part of the network covers 407 kilometers and includes 178 vehicles that execute 2,814 trips per operational day.

Figure 2 depicts an overview of the south-western portion of the network. The complete network is described in detail in [37].

4 Validating Operational Strategies

For the overall validation of the described approaches for operational resilience strategies three sets of simulation experiments are conducted:

1) Basic verification of bunching mitigation measures for delay management on an artificial transport network;
2) validation of synchronization assurance measures for delay management on the bi-modal public transit network of Cologne; and
3) validation of disruption management measures via a simulated disruption of Cologne's light-rail network, as described in summary in [13].

If not specified otherwise, all results are averaged over 100 simulation runs, with statistics being collected – after a stabilization phase – between 8am and 6pm of simulation time.

4.1 Verification of Bunching Mitigation Measures

The anti-bunching strategy is influenced by three parameters:

1) the selection of stops defined as control points;
2) the information about vehicle deviation used for decision making; and
3) the maximum permissible departure time deviation a_{max}^{Diff}.

To verify the approach, these parameters are systematically varied and the resulting combinations are compared with each other and with the null case that uses no bunching mitigation strategy. The observed bunching effects as well as the average and maximum delay, earliness, and waiting time measured over the stops of the network are used as key performance indicators.

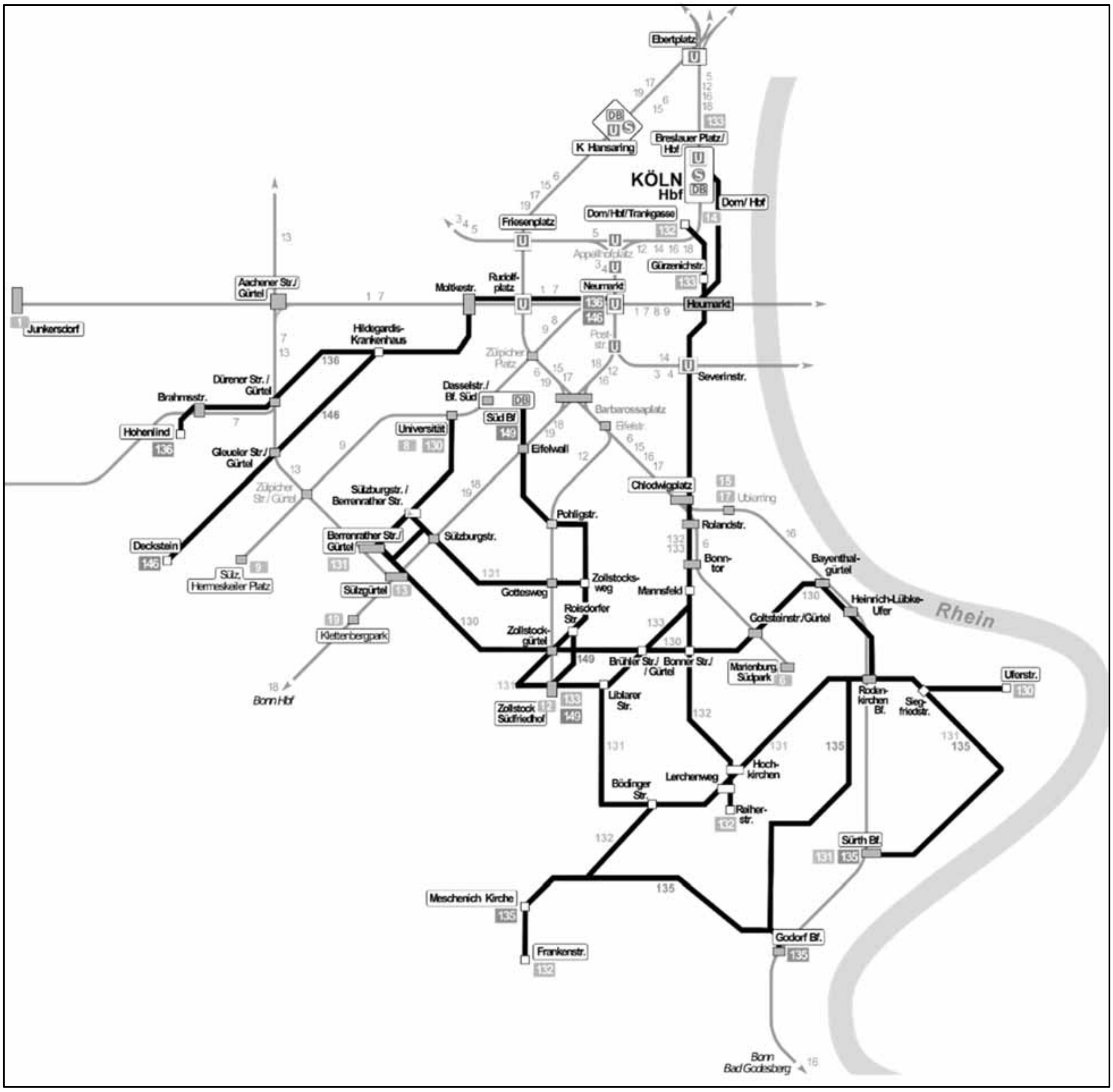


Figure 2: The southwestern part of Cologne's combined bus and light-rail network (Source: [37]).

Due to the complexity of the Cologne network and the combinatorial impracticability of comparing all potentially possible combinations of the relevant parameters, the anti-bunching strategy is verified via simulation runs on the artificial transit network first described in [26].

For this experiment, we decide on using only the starting stops of the individual lines as control points, which translates to applying no correcting measures during the vehicles' trips, versus using all stops as control points.

The threshold value of the departure time deviation a_{max}^{Diff} is varied in ten-second steps between zero and 30 seconds. Additionally, a larger threshold of 60 seconds is investigated. Together with the two options for the information to be used for decision-making (only local deviation or also environmental information, i.e., the punctuality at the next stop), this results in 20 different variations of the anti-bunching strategy. Key results are shown in Table 1.

The comparison of the observed bunching effects verifies the anti-bunching strategy and shows expected patterns: The application of any version of anti-bunching measures leads to a reduction of the fraction of diminished safety distances compared to the null case. The reduction decreases with increasing value of a_{max}^{Diff} and is stronger when all stops act as control points. The former can be explained by the fact that with increasing a_{max}^{Diff} the departure time deviation is no longer limited by this threshold value but by the travel and stop times determined on the basis of the simulation parameters or empirical data, i.e. it approaches the uncontrolled case.

The key performance indicators for delay, earliness, and waiting time from Table 1 confirm the observed patterns. The indicators were calculated by determining the average and maximum delay/earliness/waiting time at each stop of the network over the departures taking place there. These values were then averaged over the simulation runs and stops performed on the network.

As expected, the average and maximum delay increases with decreasing a_{max}^{Diff}, since vehicles are no longer able to form time buffers and thus absorb potential delays. At the same time, earliness and waiting time decrease. It is also plausible that the maximum waiting time is higher when only selected stops are used as control points, since vehicles between them accumulate larger time buffers than when all stops act as control points. Since the time buffers introduced by vehicles in front of control points are directly converted into additional waiting time, the maximum waiting time thus increases.

4.2 Validation of Synchronization Assurance Measures

To validate the synchronization assurance strategies, two different timetables for the bi-modal public transit network of Cologne were generated using the approach from [33] and examined with regard to their suitability for the implementation of directed transfer connections between the two modes of transport. Departing bus vehicles are to wait at selected control points for arriving rail vehicles to allow transferring passengers quickly.

One timetable, designated KVB-BT+, represents the optimal solution for the service regularity of the overall system. The other timetable, denoted KVB-BT-C+, is generated taking into account the transfer specifications and a balanced weighting between service regularity and fulfillment of the specifications.

Establishing transfer connections at all stops of a network is neither possible nor practical. Ideally, the most relevant interchanges should be identified based on information about passenger trips, such as origin-destination matrices, as well as operational and policy considerations, and the overall timetable should be developed with this information in mind. The most interesting potential transfer nodes of the Cologne network are the bus stations Chorweiler, Ostheim, and Porz Markt. The three selected interchanges have dedicated stops for all departing bus lines and – with the exception of the Porz Markt – all departing light-rail lines. At the stop Porz Markt, the light rail lines 7-T01 and 8-T01 as well as 7-T02 and 8-T02 share common stops, but since the variants of line group 8 share a significant part of their route with the corresponding variants of line group 7 and the latter have significantly longer routes, it can be assumed that lines 7-T01 and 7-T02 are more relevant for the establishment of interchange connections.

The three selected stops also represent a cross-section of different possible constellations of start/end or transit stops: The Chorweiler stop is the final/starting stop of the rail lines 18-T01/T02 as well as the bus line 126-B01/B02. In addition, bus lines 120-B01/B02, 121-B01/B02 and 125-B01/B02 leave from here. The Ostheim stop, on the other hand, is a transit stop for the train lines 9-T01/T02 and the bus lines 152-B01/B02, as well as the start/end stop for the bus lines 157-B01/B02. Finally, the stop Porz Markt is the start/end stop for the bus lines 152-B01/B02, 154-B01/B02, 160-B01/B02, 161-B01/B02, 162-B01/B02 and the train lines 8-T02/T01, as well as a through stop for the train lines 7-T01/7-T02.

For the light rail network, we assume a boarding time of three seconds per passenger and a departure interval of ten minutes, the arrival rates of all stops are set such that $T_{f,s}^{Rail} = 20$ seconds. In addition, all stops in the rail network act as control points at which on-time departures are enforced.

An arrival time of three seconds per passenger is also assumed for the bus network and the arrival rates of all stops are set so that $T_{f,s}^{Bus} = 20$ seconds. The connections of the Cologne bus network have an average

planned travel time of about 94 seconds. Accordingly, $\gamma^{Bus} = 1 - (\frac{20}{94}) \approx 0.79$ is set. Since individual traffic is subject to stronger fluctuations in travel time than rail traffic, $\eta^{Bus} = 0.21$ is set following a random sample sensitivity analysis.

Strategy			Delay [sec.]		Earliness [sec.]		Waiting time [sec.]	
Control points	Information	a_{max}^{Diff}	∅	Max.	∅	Max.	∅	Max.
-	-	-	4.4	84.6	14.1	104.8	0.0	0.0
Start	Local	0	8.5	95.6	4.6	61.7	1.5	57.0
Start	Local	10	4.6	85.6	10.7	71.4	0.4	58.2
Start	Local	20	4.4	84.3	12.5	81.3	0.2	55.3
Start	Local	30	4.4	84.1	13.5	88.5	0.1	44.9
Start	Local	60	4.4	82.3	14.1	104.0	0.0	16.9
All	Local	0	10.4	98.4	0.0	0.0	2.6	27.9
All	Local	10	5.4	89.6	5.0	10.0	1.5	27.6
All	Local	20	4.4	82.8	8.8	20.0	0.8	26.7
All	Local	30	4.3	85.4	11.1	30.0	0.5	25.3
All	Local	60	4.4	84.7	13.7	60.0	0.1	20.5
Start	Environment	0	7.1	98.4	5.6	62.4	1.3	58.4
Start	Environment	10	4.6	86.9	10.8	69.9	0.5	57.8
Start	Environment	20	4.3	84.8	12.6	81.0	0.2	56.4
Start	Environment	30	4.3	82.8	13.5	90.3	0.1	47.1
Start	Environment	60	4.4	86.4	14.1	104.3	0.0	15.2
All	Environment	0	8.4	100.4	0.7	13.4	2.4	27.7
All	Environment	10	5.3	89.7	5.2	15.2	1.5	27.6
All	Environment	20	4.4	84.6	8.8	20.0	0.9	26.7
All	Environment	30	4.3	84.6	11.2	30.0	0.5	24.9
All	Environment	60	4.4	83.8	13.7	60.0	0.1	20.4

Table 1: Average delay, earliness, and waiting time of different management strategies; source: [37].

In addition, slight variations in travel time due to traffic signals are mapped – again, following a sensitivity analysis – by setting $\iota = 0.01$, i.e. per traffic signal on a connection the standard deviation of travel time is increased by one percent. Finally, the first stops of all lines as well as the stops that represent connection points to long-distance traffic (i.e., 128 of the 1.242 stops of the bus network) are defined as control points. At these control points, the previously identified most appropriate anti-bunching strategy is implemented: Vehicles are allowed to depart, based on local information, a maximum of ten seconds before their scheduled departure time.

Under these parameters, the suitability of both schedules for the implementation of synchronization assurance measures is tested by means of six experiments. For this purpose, we systematically vary the connection waiting time wt_c and the transfer time t_t to comply with the transfer connections defined for the timetable and the results are compared, as far as possible, with key performance indicators of the base case without transfer connections (marked by the parameter values "-").

The waiting time wt_c is varied in 60-second increments from 0 to 180, since the traffic planning specifications under which schedule KVB-BT-C+ was generated allow bus vehicles to depart as scheduled up to three minutes before the feeder vehicle of the light-rail service. In order to maintain the transfer connection, they must therefore wait up to three minutes for the light-rail vehicle if they arrive on time. In addition, a shorter waiting time of 30 seconds is also tested.

Values of 0 and 30 seconds are used for the transfer time t_t, which indicated how long vehicles wait after completing a transfer connection until their actual departure.

The former is only used together with a waiting time of $wt_c = 0$ to capture the inherent suitability of the schedules to implement transfer connections, i.e. without (significant) intervention of the dispatcher. The value of 30 seconds for t_t was chosen so that any (artificial) delays due to this value would not affect the simulation metrics too much. In the real system, waiting buses would start as soon as all transferring passengers have entered the vehicle. Based on passenger numbers, a plausible (average) value is chosen that should allow the majority of passengers to transfer comfortably. The time t_q, which bus vehicles wait until they are asked again whether the transfer connection is fulfilled, is set to 30 seconds uniformly for all experiments.

Table 2 lists the resulting percentage of connections made by the different bus lines at the three stops during the measurement period under the different timetables and parameters. These are the departures of the bus lines during which the relevant feeder vehicles of the light-rail lines arrive within the specified time interval.

The results show that schedule KVB-BT-C+, as expected, is already better suited for the transfer of transfer connections than schedule KVB-BT+ without significant intervention by the dispatcher.

Under the latter, inherently only an average of 14.05 percent of all potential transfer connections are fulfilled, while under schedule KVB-BT-C+ an average of 42.20 percent of all transfer connections are fulfilled.

If the dispatcher is prompted to intervene more strongly by increasing the maximum waiting time wt_c, the proportion of fulfilled transfer connections can be increased to an average of up to 29.63 percent under schedule KVB-BT+. Under timetable KVB-BT-C+, on the other hand, the average share of fulfilled transfer connections can be increased to up to 87.49 percent. This effect is to be expected because under the KVB-BT-C+ timetable the bus lines are forced to depart from the three interchanges within three minutes of the departure times of the light-rail lines concerned.

By deliberately delaying the vehicles of the bus lines by up to three minutes during the simulation, the probability that the interchange connections will be fulfilled in regular operation must therefore be increased. Under the KVB-BT+ timetable, however, the application of this strategy does not have such a pronounced effect since the synchronizations between the departures of the bus and light-rail lines are largely random without taking traffic planning requirements into account.

Time table		KVB-BT+					KVB-BT-C+				
	wt_c [sec.]	0	30	60	120	180	0	30	60	120	180
	t_t [sec.]	0	30	30	30	30	0	30	30	30	30
Chorweiler	120-B01	0.00	0.00	0.00	0.00	0.00	0.00	97.38	**98.33**	**98.33**	**98.33**
	120-B02	0.77	0.72	0.92	0.82	0.73	63.37	63.48	63.78	63.48	**64.32**
	121-B01	0.00	0.00	0.00	0.00	0.00	69.97	**70.53**	68.56	68.30	68.23
	121-B02	0.00	0.00	0.00	0.00	0.00	0.07	97.38	**98.33**	**98.33**	**98.33**
	125-B01	13.44	12.57	12.05	98.28	98.06	0.00	0.00	0.00	0.00	**98.32**
	125-B02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	**98.33**	**98.33**
	126-B02	96.67	96.67	96.67	96.67	96.67	0.00	0.00	0.02	**98.33**	**98.33**
Ostheim	152-B01	49.88	48.91	37.50	0.88	0.00	**99.95**	98.78	49.13	46.86	5.22
	152-B02	0.00	0.00	0.00	0.00	0.00	99.98	**100.00**	99.99	**100.00**	**100.00**
	157-B01	0.00	0.00	0.00	49.92	49.03	99.70	**99.80**	99.73	95.96	83.00
Porz Markt	152-B01	50.00	50.00	50.00	50.00	50.00	0.00	26.26	**100.00**	**100.00**	**100.00**
	154-B01	0.00	0.00	0.00	0.00	50.00	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	160-B01	0.00	30.36	50.00	50.00	50.00	0.00	0.00	0.00	**100.00**	**100.00**
	161-B01	0.00	0.00	0.00	0.00	0,00	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
	162-B01	0.00	0.00	0.00	0.00	50.00	0.00	0.00	0.00	0.00	**100.00**
	Ø	14.05	15.95	16.48	23.10	29.63	42.20	56.91	58.53	77.86	**87.49**

Table 2: Average ratio of kept transfer connections at core stops examined with different combinations of timetables and transfer times; source: [37].

The fact that even under the KVB-BT-C+ timetable not all transfer connections can be fulfilled can also be explained by the fluctuating travel and stop times due to the dynamic conditions.

However, the effects induced by the synchronization strategy are not limited to the interchanges and the directly affected lines. Delays and associated bunching effects are often carried over the entire remaining route and are also transferred to the reverse directions of the line groups, in some cases in the form of increased delay and/or earliness.

4.3 Validation of Disruption Management Measures

To validate disruption management measures, a disruption in the Cologne light-rail network is simulated (as already briefly discussed in [26]). All simulation parameters described in the previous section are retained, as is the stopping strategy used, where early *vehicles* are forced to depart on time at each stop.

The simulated disruption is the one-sided blockade of the connection between the Neumarkt (NEU) and Heumarkt (HEU) stops in the city center between nine and ten in the morning. As a result, no operations can take place on the connection during this period.

Without explicitly addressing the disruption, services are extremely disrupted in their operations, with average delays of between about one and two hours. If, on the other hand, the disruption is responded to by splitting the route, its effects cannot be completely eliminated, but they have a much more moderate impact on the operating schedule, with line delays of no more than approx. 44 seconds.

Figures 3 and 4 show as an example the departure times of all simulated trips of the line 1-T01 during the of the line 1-T01 at all stops of the route during the measurement period are plotted. First of all, before the onset of the disruption (upper dashed line), the behavior is the same both with and without incident management: Vehicles are able to travel without further complications from their starting stop at Junkersdorf (platform #1243) to their final stop at Bensberg (platform #1274). As soon as the fault becomes active, however, the resulting system behavior differs however, differs significantly.

Without incident management, all trips that reach the Neumarkt stop (platform #1254) during the incident are delayed there until the incident is cleared (lower dashed line).

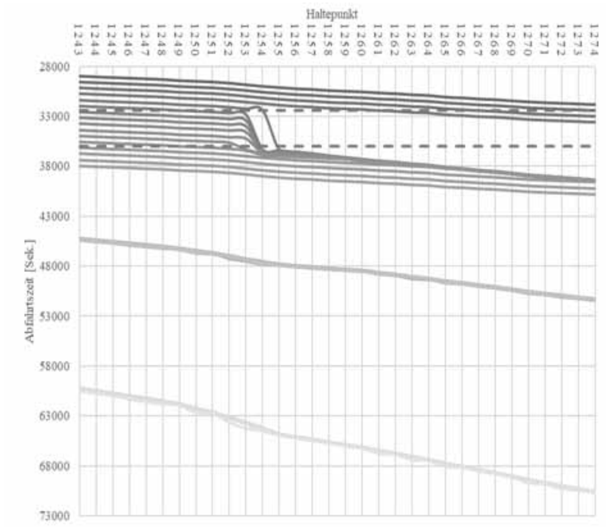


Figure 3: Impact of a 60-minute outage on a light-rail line without mitigation strategy: Most vehicles servicing the line wait at the blocked location for the outage to be repaired; after that the vehicle bunching effect [14] causes the service to be unusable for the rest of the operational day. (Source: [37]).

The vehicles then continue their journeys as a convoy, without even spacing. Since the vehicles cannot make up for their delay, also due to the unintentional column movement, it is transferred to the subsequent journeys in the opposite direction, where the same phenomenon can be observed. Without external intervention, this effect cannot be broken, so that the regular service breaks down and only sporadically a single column of delayed trains serves the stops.

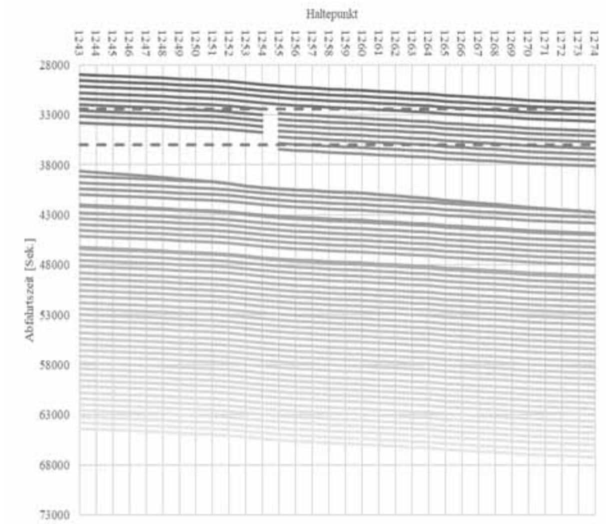


Figure 4: Impact of a 60-minute outage on a light-rail line with an effective mitigation strategy: The service level bounces back to almost normal half an hour after the outage has been repaired. (Source: [37]).

With disruption management, however, regular operations can be maintained to a large extent. As shown in Figure 4, the trips that would reach the Neumarkt stop during the disruption are shortened at this stop. The service of the stops behind the disruption, on the other hand, is taken over by the vehicles that travel in the opposite direction only to the stop Heumarkt (platform #1255). After the end of the disruption and after the vehicles have arrived at the Junkersdorf stop by means of a regular trip in the opposite direction, regular service is resumed.

5 Conclusion

This paper described the application of a bi-modal, mesoscopic simulation model in the test and evaluation process of measures and strategies aimed at increasing the resilience of urban transit networks against the impact of climate change. The paper provided an introduction to transit resilience as well as delay management and disruption management measures, described urban transit modeling, simulation, and timetable generation. As a main part, the paper examined the evaluation of measures and strategies based on a simulation model of the urban transit network of the city of Cologne, Germany.

As described, the research indicate that the delay and disruption management strategies increase the resilience of transit networks as expected. Additionally, the results also indicate that the described evaluation process indeed is applicable to test and evaluate both pre-planned and emergency response strategies.

In further research steps, the described strategies will first be refined using the model of another urban transit network, and then carefully transitioned to the application in a real-world transit network.

References

[1] Olfert A, et al. 2021. *Sustainability and resilience - a practical approach to assessing sustainability of infrastructure in the context of climate change.* G. Hutter, M. Neubert and R. Ortlepp. Building resilience to natural hazards in the context of climate change - Knowledge integration, implementation, and learning. Springer, 2021.

[2] Rossetti MA. Potential impacts of climate change on railroads, In: *Workshop on the Potential Impacts of Climate Change on Transportation*, Washington, D.C., October 1-2, 2002.

[3] Nemry F, Demirel H. *Impacts of Climate Change on Transport: A focus on road and trail transport infrastructures*, 2012 Joint Research Centre Institute for Prospective Technological Studies, JRC72217, EUR 25553 EN, 93 pg., 2012.

[4] Folke D. Resilience: The emergence of a perspective for social–ecological systems analyses. *Global Environmental Change 2006*; 16(3): 253-267.

[5] Seelinger L, Turok I. Towards Sustainable Cities: Extending Resilience with Insights from Vulnerability and Transition Theory. *Sustainability* 2013, 5(5): 2108-2128.

[6] Doorn N. Resilience indicators: opportunities for including distributive justice concerns in disaster management. *Journal of Risk Research* 2017; 20(6): 711-731.

[7] Holling C. Engineering resilience versus ecological resilience. In: *Engineering Within Ecological Constraints*, P. Schulze, Ed., Washington D.C., National Academy Press, 1996, pp. 31-44.

[8] Ullrich O, Lückerath D, Speckenmeyer,E. Do regular timetables help to reduce delays in tram networks? – It depends! *Public Transport 2016,* 8: 39-56.

[9] Forzieri G, et al. Escalating impacts of climate extremes in critical infrastructures in Europe. *Global Environmental Change* 2018, 48.

[10] Forzieri G. et al. *Resilience of large investments and critical infrastructure in Euope to climate change.* JRC Report, 2016.

[11] Milde K, Lückerath D, Ullrich O. *ARCH Disaster Risk Management Framework*. EU H2020 ARCH (GA No. 820,999), Deliverable D7.3, 2021.

[12] *City Resilience Development - Guide to combine disaster risk management and climate change adaptation – Historic areas*, CWA 17727:2022, European Committee for Standardization, Brussels, 2022.

[13] Lückerath D, Bogen M, Rome E, Sojeva B, Ullrich O, Worst R, Xie J. Strategies to Mitigate the Impacts of Climate Change Related Events on Public Transit Networks. *Proc. 24th Symposium Simulationstechnik (ASIM 2018)*, Hamburg, Germany, October 03-04, 2018, 175-182.

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

[15] Yu B, Yang Z. A dynamic holding strategy in public transit systems with real-time information. *Applied Intelligence*, Vol. 31, No. 1, S. 69-80, 2009.

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

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

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

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

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

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

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

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

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

[25] Ullrich O, Lückerath D. A Bi-Modal Simulation Model to Increase the Resilience of Public Transit Networks. *Simulation Notes Europe (SNE)* 2022; 32(2): 103-112.

[26] Lückerath D, Rishe N, Speckenmeyer E, Ullrich O. A Mesoscopic Bus Transit Simulation Model Based on Scarce Data. *Simulation Notes Europe (SNE)* 2018; 28(1): 1-10. DOI: 10.11128/sne.28.tn.10401

[27] Ullrich O, Lückerath D. An Introduction to Discrete-Event Modeling and Simulation. *Simulation Notes Europe (SNE)* 2017; 27(1): 9-16 DOI: 10.11128/sne.27.on.10362

[28] Lückerath D, Ullrich O, Kupicha A, Speckenmeyer E. Multi-depot multi-vehicle-type vehicle scheduling for Cologne's tram network. In: Proc. *ASIM-Workshop STS/GMMS 2014*, ARGESIM Report 42, ASIM-Mitteilung AM 149, ARGESIM/ASIM Pub., TU Vienna/Austria, 191-197.

[29] Lückerath D, Ullrich O, Speckenmeyer E. Applicability of rescheduling strategies in tram networks. In: Proc. *ASIM-Workshop STS/GMMS 2013*, ARGESIM Report 41, ASIM-Mitteilung 145, ARGESIM/ASIM Pub. TU Vienna, 2013, 57-64.

[30] Desaulniers G, Hickman M. *Public transit, Handbooks in Operations Research and Management Science,* Barnhart, C., Laporte, G. *(eds.), North Holland,* Amsterdam 2007, 69–127.

[31] Bampas E, *et al. Periodic metro scheduling. Proc. 6th Workshop on ATMOS, Dagstuhl, Germany, 2006, 63–77.*

[32] Ibarra-Rojas OJ, López-Irarragorri F, Rios-Soli, YA. Multiperiod synchronization bus timetabling, *Transport Sci. 2015, 50:* 805–822.

[33] Lückerath D, Rishe N, Speckenmeyer E, Ullrich O. A disjunctive program formulation to generate regular public transit timetables adhering to prioritized planning requirements. *Networks 2018*, 72(2): 217-237.

[34] Vossloh Kiepe GmbH. *Elektrische Ausrüstung der Niederflur-Stadtbahnwagen K4000 der Kölner Verkehrs-Betriebe AG 2003.* Druckschrift Nr. 00KV7DE.

[35] Vossloh Kiepe GmbH. *Elektrische Ausrüstung des Niederflur-Stadtbahnwagens K4500 für die Kölner Verkehrs-Betriebe AG 2003.* Druckschrift Nr. 00KN2DE.

[36] Vossloh Kiepe GmbH. *Elektrische Ausrüstung der Hochflur-Stadtbahnwagen K5000 der Kölner Verkehrs-Betriebe AG 2003*, Druckschrift Nr. 00KB5DE.

[37] Lückerath D. *Ein Simulationsmodell für Öffentlichen Personennahverkehr mit regelbasiertem Verkehrs-managent* (Dissertation). Institut für Informatik, Universität zu Köln; 2017, 182 pg.

Towards a Scenario Toolkit for Autonomous Systems

Ahmad Naja[1*], Siddhartha Gupta[1], Umut Durak[2], Sven Hartmann[1]

[1]Institute for Computer Science, TU Clausthal, Clausthal-Zellerfeld, Germany; * *ahmad.naja@tu-clausthal.de*

[2]German Aerospace Center (DLR), Braunschweig, Germany; *umut.durak@dlr.de*

SNE 34(1), 2024, 13-21, DOI: 10.11128/sne.34.tn.10672
Selected ASIM WS 2023 Postconf. Publication: 2023-10-15
Received Revised Improved: 2024-02-12; Accepted: 2024-02-18
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Scenario-based approaches have recently been widely adopted in the automotive and aviation industries. They aim to define and manage the test cases in a better way, thereby significantly reducing the risks and defining the safety argument for the system. To improve and complement the safety conditions, the Operational Design Domain (OOD) defines the operational boundaries of a driving automation system to specify the scope of the safety case, represented by human and machine-readable languages. Many approaches and standards that involve the essential modelling and safety concepts have been introduced to represent scenario-based simulation. Some parts of these approaches were implemented as graphical tools, such as System Entity Structure (SES) and Pruned Entity Structure (PES) tools, which are based on ontology and its derived structure. However, an extensive implementation covering scenario modelling and management, ODD, and assessment is still missing. This paper proposes an adapted scenario-based approach based on related research, and implements it in a robust GUI tool called Operational Domain Modeling Environment (ODME). ODME is progressing to fill the gap and address the aforementioned limitations by covering the modelling functions and safety approaches in one comprehensive environment with a wide range of capabilities and features.

Introduction

The subject of modeling aims to represent a certain aspect of reality for a particular purpose. Systems, processes, and phenomena can all be represented by models.

Afterward, the system behaviors can be generated through the simulation process using the model [1]. Creating and developing models is considered as an initial step to simulate and test the system based on different use cases, called scenarios.

A scenario describes a system's behavior based on its operating conditions and situations, changing of its parameters through the time, and the mutual interaction of its components with each other [2]. Scenario modelling helps researchers better plan and lower risks by exploring a variety of prospective outcomes, comparing them using specific standard metrics, and testing decisions [i].

Developing a scenario goes through significant steps, starting with defining the scenario by the stakeholders and finishing by generating the necessary executable specifications. Scenarios as executable specifications are input to the simulation environment [3].

One of the biggest problems facing the car industry is ensuring the safety of autonomous vehicles. Scenario-based development and test methods are potential methods for testing and validating autonomous driving capabilities [4].

The scenario-generating process of the scenario-based approach carries forward a safety argument, which is crucial for the system's release. So, the scenarios must be created and documented methodically. In addition, they have to be traceable throughout the development process [4]. Technology firms and automakers use the Operational Design Domain (ODD) concept to specify the safe operating conditions for their Automated Driving Systems (ADS). An ODD establishes where the ADS are intended to function correctly by definition [5]. Therefore, scenarios and OOD complement each other to provide a safety argument for an autonomous vehicle.

[i] https://www.synario.com/scenario-modeling-what-you-need-to-know/

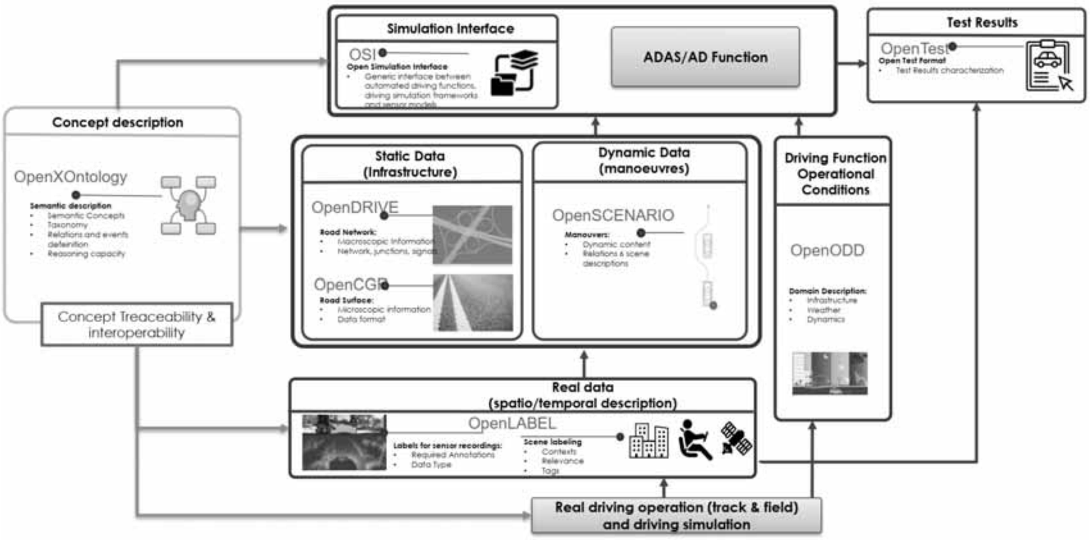


Figure 1: Simulation Process Workflow Based on a Family of ASAM Standards [6].

Organizations and researchers develop many standards to represent the scenario computationally, such as OpenSCENARIO from the Association for Standardization of Automation and Measuring Systems (ASAM). ASAM played a lead role in setting standards that cover the whole simulation pipeline. Figure 1 shows a simulation process workflow based on a family of ASAM standards, which illustrates the different steps a scenario goes through, beginning with a concept description and ending with testing results. OpenSCENARIO is the point of interest used for driving simulation and virtual development, testing and validation of driving assistance functions, automated and autonomous driving [ii].

There are some concerns with the approach illustrated in Figure 1. Many standards are still under development and have not been shown to be compatible with each other. Many third-party tools have been shown to use one of these standards, mostly focusing on OpenSCENARIO, but none can handle the complete workflow. A more integrated and compact workflow is needed along with its associated tool-set.

This paper provides new insights to define an adapted approach based on the ASAM simulation workflow in Figure 1 and other research work. This adapted approach will compact the simulation process based on scenario modelling, scenario generation, management, and ODD definition.

[ii] `www.asam.net/standards/detail/openscenario/`

Furthermore, a modelling environment that implements the new approach will be introduced, called the Operational Domain Modeling Environment (ODME). The presented development method utilizes System Entity Structures (SES) and metamodeling that provide the capability to develop models and generate executable software entities with the corresponding model transformations within the technical spaces [7]. The SES and its related project, Pruned Entity Structures (PES) tools presented in [8], were a starting point for achieving the goals. After refactoring, they will be ready to expand and implement the scenario-based approach.

The paper begins by giving a background in section I about SES, scenario representation, Operational Design Domain, ASAM Standards, and SES and PES Tools. In section II, the newly suggested approach will be introduced and illustrated. Section III will implement the theoretical approach in a practical modelling environment, and section IV gives a short conclusion and shows some recommended views for future work.

1 Background

1.1 System Entity Structure (SES)

SES is described as a framework for knowledge representation of system coupling, decomposition, and taxonomy [9] and is considered as an enhancement in the discipline of system theory-based approaches for modelling and simulation [10].

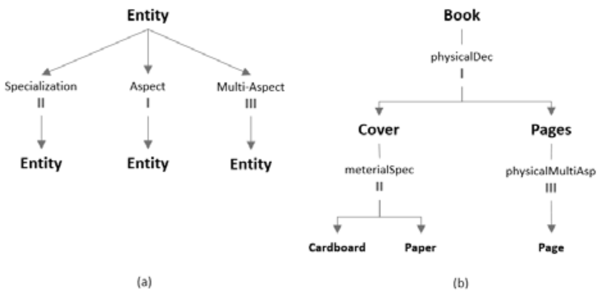


Figure 2: (a) SES Nodes and Relationships, [11] and (b) an Example[12].

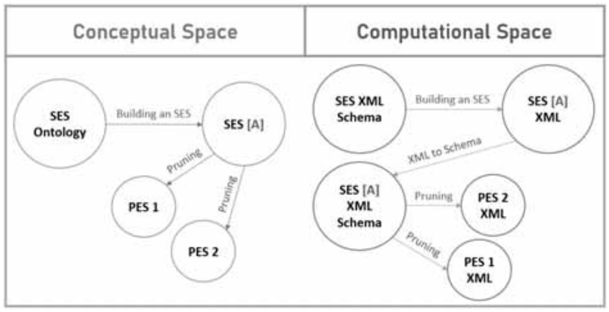


Figure 3: Computational Representation [16].

Utilizing interactions between decomposition, coupling, and taxonomies allows for the succinct specification of a family of models [13]. SES is further described as a formal ontology framework specifying a system's components and hierarchical relationships [11]. Figure 2 shows SES nodes and relationships and a simple SES tree example.

An SES is represented as a labelled tree. The entity, Aspect, Specialization, and Multi Aspect are the four different types of nodes [14].

- **Entity:** It depicts the system's elements that are artificial or real. It is an object of interest that may have variables linked to it. Other node types are utilized to describe parent and child entities.
- **Aspect:** It indicates the decomposition relationship of an Entity node. It represents processes for breaking down larger objects into more fine-grained ones.
- **Specialization:** It represents an Entity's taxonomy. Specialization denotes groups or families of distinct forms that an object may take.
- **Multi Aspect:** This particular type of aspect describes a multiplicity connection and shows that the parent entity is composed of many entities of the same type.

Several axioms define the SES as well, which are: inheritance, valid brothers, strict hierarchy, uniformity, attached variables, and alternating mode [15].

Given its foundations in the theory of modelling and simulation and its expressive strength and clarity with only a few axioms, SES is suggested as the foundation of the proposed intermediate metamodel in the simulation model package.

Thus, SES is appropriate as a straightforward intermediate metamodel that establishes a formal framework that is clear and understandable [17].

1.2 Scenario Computational Representation

The machine-readable format for SES and Pruned Entity Structure (PES) is called computational representation. There are two significant operations in the conceptual space, as shown in Figure 3.

An SES Ontology- a specific SES - is constructed utilizing the constructs, structure, and axioms of SES. The pruning process generates the Pruned Entity Structures (PESs) from this specific SES. According to [18], we may express the SES ontology with an XML Schema and the specific SES as an XML file in the computational space. Then, they proposed creating an XML that specifies a particular SES to an XML Schema. The construction and validation of PESs during pruning finally employ this schema. PESs ultimately become XML instances.

The schema for the SES ontology could be defined using the XML Schema Definition Language (XSD). An XML document's restrictions and structure may be described using XSD [19].

1.3 Operational Design Domain (ODD)

SAE J3016 defines the Operational Design Domain (ODD) for a driving automation system as *"Operating conditions under which a given driving automation sys-tem, or feature thereof, is specifically designed to func-tion, including, but not limited to, environmental, geo-graphical, and time-of-day restrictions, and the requi-site presence or absence of certain traffic or roadway characteristics."*

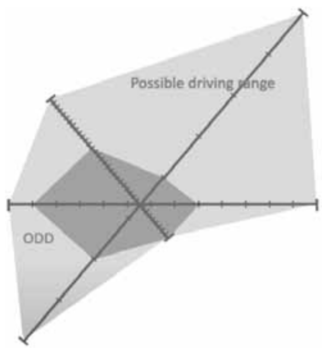


Figure 4: Operational Design Domain (ODD) [20].

In short, the ODD establishes the limits that the driving automation system is intended to work within and, as a result, will only function when these criteria are met, as shown in Figure 4.

The ODD restricts where the automated driving system (ADS) is valid, which helps to define the scope of the safety case and the verification. Use cases are required to give a strategy for a set of operating conditions (OCs) and verify that it remains within the range of the ODD [iii].

ODD can control the coverage of scenarios and provide a list of operation conditions, which can be used later as an effective input for the system assessment process. Some requirements are important when ODD is defined. For instance, ODD should be generated in a format understandable for humans, such as tables. On the other hand, it has to be machine-readable, such as an XML file.

1.4 ASAM Standards

In the automobile sector, development and test toolchains can be typically built to ASAM standards, which are public specifications and can be optionally used in development. To determine the Standard-compliance of products, ASAM advises and promotes best practices. Throughout vehicle development, ASAM standards specify interfaces, protocols, file formats, and data models.

ASAM-based tools and products provide seamless data exchange and simple integration into already-existing value chains [iv].

[iii] www.claytex.com/tech-blog/
[iv] https://www.asam.net/standards/standard-compliance/

ASAM provides a family of standards for the simulation domain, which have repeatedly proven themselves in various development processes. These standards must interact to generate a global view of the simulation process. In the following, some related ASAM standards are mentioned [21]:

- **OpenXOntology:** An ontology-based framework for notions like roads, lanes, and traffic participants is provided by OpenXOntology. The ASAM Open XOntology comprises several interconnected components, including core, domain, and application ontologies.
- **OpenODD:** For connected autonomous cars (CAVs), OpenODD seeks to create a format that may describe a specified Operational Design Domain. The ODD defines the functional requirements for connected autonomous vehicles and outlines the environmental characteristics that CAVs must be able to control.
- **OpenSCENARIO:** The dynamic content of the world is defined by OpenSCENARIO, for instance, the anticipated behaviour of traffic participants and how they should interact with one another and their environment.
- **OpenDRIVE:** The primary goal of OpenDRIVE is to offer a description of the road network that can be used as input into simulations to create and verify advanced driver assistance systems (ADAs) and autonomous driving features.

1.5 SES and PES Tools

The initial scenario workflow using SES [22] was implemented by SES tool, whereas, the pruning capabilities were provided by PES Tool. SES and PES projects added the outlines for a formal strategy to create a scenario specification language [23]. Figure 5 demonstrates basically the workflow of SES-PES projects.

In SES Tool, The user may access a wide variety of widgets using the graphical user interface that helps with modelling. It uses a collection of elements and axioms to describe knowledge about system connection, taxonomy, and decomposition [24]. While the SES model developed is pruned using PES Tool, several scenarios are produced. Pruning is a technique that creates a distinct system structure from a domain model; the outcome is known as a Pruned Entity Structure (PES).

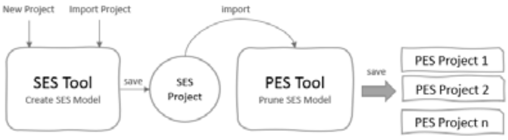


Figure 5: SES and PES Projects Structure.

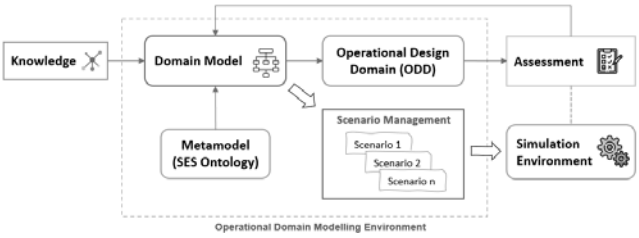


Figure 6: Proposed Scenario-based Approach Workflow.

An SES Model represents a family of models for a certain application domain. All of a system's potential configurations are taken into account when using SES modelling. An SES model's tree has to be trimmed to get a specific configuration. To achieve this configuration, a domain model's tree that is a selection-free tree, pruning removes extraneous structure based on the definition of a realistic frame.

A domain model is often reduced by pruning by eliminating options for an entity with numerous attributes and specializations made up of several entities. A domain model's tree may be pruned by giving the values of the variables, choosing one entity from the available specialization node possibilities, and indicating the cardinality in a Multi Aspect node [8].

Although the tool-set provided the modelling and simulation community with many advantages, it is currently not scalable and requires several enhancements. Using two tools and managing the pruned models was a critical problem that made the project complicated and hard to use.

2 General Scenario-based Approach for Autonomous Systems

The proposed scenario-based approach uses the fundamental elements of many works involving scenarios such as the ASAM workflow [6] and other research works like [25], [26], integrating many features of scenario simulation principles.

The simulation model process in the proposed approach goes through a sequence of fundamental steps, from knowledge acquisition to finding the best scenario to be executed. The domain model should be created depending on some knowledge generated by experts or raw data. A domain model defines an abstract representation of all the elements of the simulated system, so it needs to be pruned to produce a particular use case or scenario. Scenario Manager aims to organize the created scenarios and their attributes and help to arrange scenarios based on different metrics. Afterward, the selected scenario will be executed physically or virtually to be tested and assessed. Finally, the metrics will be evaluated to send feedback to the domain model to be improved. Figure 6 illustrates the suggested scenario-based approach.

Knowledge Source. Common methods to generate valuable scenarios are using the domain experts' knowledge and creating the scenarios manually. Data from real driving situations could also be an input for scenario modelling (not the focus of our current approach).

Metamodel (SES Ontology). An ontology offers a vocabulary for a specific domain by computerizing the specification of the meaning of definitions and describing the concepts and relationships that are significant in a given domain. As ontologies explain domain relationships and entities in a simple and machine-interpretable way, they serve as a bridge between humans and computer systems. Model transformations are provided as an automated way to create an executable scenario definition, whereas metamodeling is suggested as a method to create a graphical modeling language. SES is used for metamodeling to incorporate all the elements of a scenario that could be simulated in autonomous vehicles [23]. Based on this metamodel, a scenario modeling approach is developed.

Domain Modelling. In this step, the knowledge will be used to create a model based on SES ontologies that describe the system in human-readable representation. This model will be considered an abstraction of all possible scenarios, so it needs to be pruned later to generate derived models or individual scenarios. Moreover, the variables and constraints used in the model will be fundamental to defining ODD.

Scenario Management. The created domain model will be derived in different scenarios by the pruning process. However, the generated scenarios need a mechanism to be organized. The scenario manager aims to achieve this goal by developing a method to facilitate storing and managing the scenarios. Scenarios also need to have certain metrics associated with them.

Define Operational Design Domain (ODD). In this block, the operating conditions that form the system's boundary will be defined, as shown in Table 1. These operational conditions will be used as input for the assessment process.

Value Range	Aspect
Pedestrian	sporadically
Road Types	highway
Time of Day	any
Speed Range	$\leq$ 130 kph
Visibility	$\geq$ 40 m

Table 1: ODD Table Example.

Scenario Execution. Once the scenario test cases have been identified, they must be executed on real physical systems or simulation tools. During the execution, the results should be recorded and stored to be analyzed and evaluated. This step is essential to specify the safety properties and is required to develop executable specifications.

Assessment Process. Evaluating the quality and the performance characteristics, such as safety and efficiency, is crucial in developing autonomous systems. In this block, the results and remarks of execution will be processed and analyzed to evaluate the scenario, hazard analysis, and risk assessment. After that, The scenario's metrics will be updated and sent as feedback to the domain modelling block.

3 Operational Domain Modeling Environment (ODME)

ODME is a robust environment that contains all functionalities the user needs to create and prune models and manage pruned scenarios via the scenario manager.

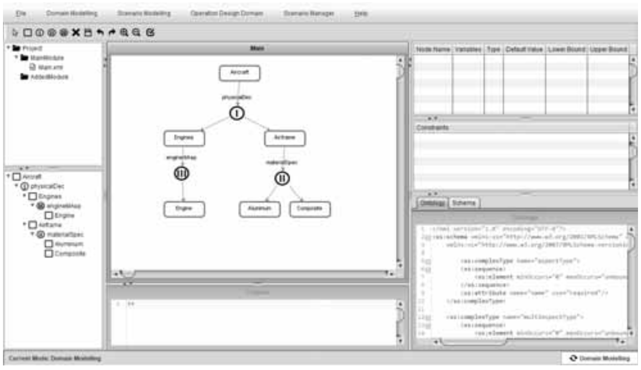


Figure 7: ODME Main Window in Domain Modelling Mode.

It is a successor to the SES tools implementing the functionalities contained in the dotted line in the proposed scenario-based approach in Figure 6. Figure 7 shows the general view of the ODME main window in domain modelling mode.

Developing the ODME project went through four important levels:

1. **Planning:** Before starting with implementation, many related projects and research were reviewed to collect ideas and create a special innovative approach. The SES and PES Tools project was also chosen as a base and starting point for ODME.

2. **Reconstruction:** two important steps were implemented in this level:

   • Refactoring Process: cleaning the code, fixing bugs, and improving the project's structure.

   • Build Mode Switch mechanism: create a smooth and simple solution to change between Domain Modelling and Scenario Modeling modes after merging SES and PES tools.

3. **Upgrade:** We added many new features to ODME, such as importing/exporting the model as a template and saving the model as a PNG file, in addition to some improvements in GUI design.
   The essential new feature in ODME is the scenario manager, which allows the user to create multi-scenario models for one domain model and delete and change the metrics of scenarios.
   The scenario files will be managed automatically and stored in separate folders that can be moved to other devices, making the created projects portable.

4. **Test and Documentation:** by using the new environment to create real use case scenarios, documenting, and suggesting more advanced features for future work.

As mentioned before, ODME consists of two modelling modes, domain and scenario modelling, as well as the scenario manager.

3.1 Domain Modelling Mode

ODME facilitates the user creating a domain model as a visual representation of real situation objects by a wide set of widgets and features. The components in the domain of the problem, and the connections between them, are represented by the Domain Model. All system entity structure components introduced in section [II] are supported. ODME added many new features which help to improve usability and make the environment reliable and user-friendly.

While ODME starts, the main window will appear in Domain Modelling mode, and a new project called Main will be generated automatically with a root node. In the Drawing Panel, the user can create a domain model using the different types of nodes in the ToolBar. While a node is connected to the graph model, the Synchronized Tree Window will be updated to show it. When the model is saved, the ontology and schema will be generated. Figure 8 shows a simple domain model created by ODME.

3.2 Scenario Modelling Mode

After creating a domain model, ODME provides a simple mode switch mechanism, which saves the domain model and forwards it to scenario modelling mode.

In this mode, the model can only be pruned to create scenarios. The nodes that can be pruned will be highlighted.

ODME support three different types of pruning [10]:

- **Multi Aspect Node Pruning** Before pruning, a Multi Aspect node must define its cardinality or the total number of aspects.
  Cardinality is assessed, and a certain number of aspects of the same kind are formed when a Multi Aspect node is pruned.
  Figure 9 shows the result after pruning the "engineMAsp" entity in Figure 8. Based on the cardinality number, which will be defined by the user (Three, for example), three engines will be generated after pruning.
- **Specialization Node Pruning** One child must be chosen to create a valid variation per the specialization requirement. Figure 8 demonstrates that the entity Airframe has two options: composite and aluminum.
  The entity that remains after pruning can, therefore, either be of type Aluminum or type composite. In Figure 9, the completed pruned entity Composite Airframe will be generated.
- **Entity variable Pruning** Entities that have variables can only be pruned. So, they will be highlighted with green in scenario modelling mode. This pruning can be achieved by updating the value of an entity's variables.
  After pruning the entity variable, the variable table will also be updated. Figure 10 shows the variable table of the "Node" entity after pruning of "Var_2".

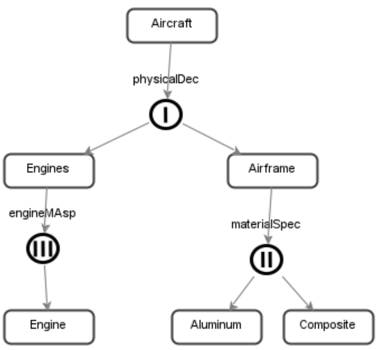


Figure 8: Domain Model Example.

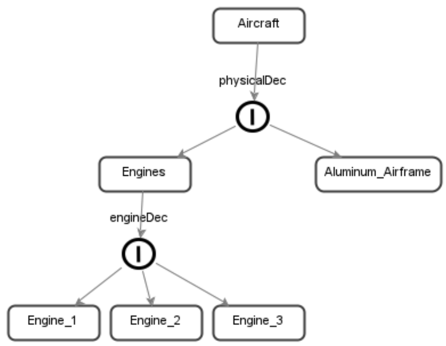


Figure 9: Scenario Model Example.

Node Name	Variables	Type	Default Val...	Lower Bou...	Upper Bou...
Node	Var_2	int	5	0	10
Node	Var_3	string	none		
Node	Var_1	boolean	none		

Figure 10: Entity Variables Table after Pruning.

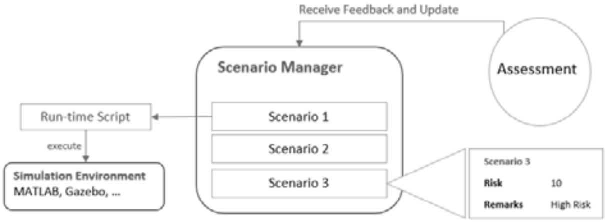


Figure 11: Scenario Manager.

3.3 Scenario Management

The domain model can be pruned to have many scenarios. The number of scenarios generated can extend to a large number, as the variability factors are many. It requires a mechanism to organize scenarios by labelling and prioritizing them before exporting and executing. Scenario manager is one of the new features in ODME used to manage the scenarios created by the tool. The following points can summarize the main goals of a scenario manager:

- Capturing all the scenarios and creating a scenarios list to organize them, in addition, to helping to access any scenario.
- Assigning criteria as classification metrics to the scenarios manually.
- Providing run-time scripts to execute the scenario directly and show the result by linking the scenarios to the simulation environment, such as Matlab and Gazebo.
- Having the possibility of a mechanism to accept feedback from the assessment and adjust the scenarios accordingly.

The last two points are still under development. Figure 11 illustrates the main functionalities of the scenario manager. Scenarios List is a simple functionality in the tool to switch between scenarios, delete scenarios and change their metrics. By double-clicking on one of the scenarios, the metrics update window will be opened, where the user can update the risk value of the scenario in addition to writing some remarks. Figure 12 shows the scenarios list.

Name	Risk	Remarks
InitScenario		
Scenario2	10	High Risk
Scenario3	2	Low Risk

Figure 12: Scenario Manager List.

4 Conclusion and Future Work

In this paper, we proposed a scenario-based approach using SES ontology, which integrates many fundamental simulation concepts, like domain modelling, scenario management, and operational design domain.

Furthermore, we implemented this approach in a robust tool called ODME. Nevertheless, ODME is still an active project open to many ideas and improvements. There are several directions for future work, aiming to implement ODD in our tool so that it could be used for the assessment process. In addition, the scenario manager needs to be optimized to compare different scenarios. The future work also includes developing a test data generator using machine learning algorithms. This data can help test a broader range of scenarios and define the operational conditions more precisely.

References

[1] P. K. Davis and R. H. Anderson. Improving the composability of dod models and simulations. pages vol. 1, no. 1, pp. 517. The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology, 2004.

[2] D. A. Kononov, V.V. Kulba, S.S. Kovalevsky, and S.A. Kosjachenko. Development of scenario spaces and the analysis of dynamics of behaviour of social and economic system. Preprint, 1999.

[3] Umut Durak, Okan Topçu, Robert Siegfried, and Halit Oğuztüzün. Scenario development: A model-driven engineering perspective. 2014.

[4] Till Menzel, Gerrit Bagschik, Leon Isensee, Andre Schomburg, and Markus Maurer. From functional to logical scenarios: Detailing a keyword-based scenario description for execution in a simulation environment. 2019.

[5] S. O.-R. A. D. Committee et al. Sae j3016. taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. tech. rep., 2016.

[6] Oihana Otaegui. Standardisation on interfaces and formats for ccam validation. connected and automated driving virtual conference, April 2021.

[7] D. Gasevic, D. Djuric, and Devedic V. Model driven engineering and ontology development. Springer, 2009.

[8] Bikash Chandra Karmokar. Application agnostic ses modeling environment and interactive pruning tool. TU Clausthal.

[9] T. Kim, C. Lee, E. Christensen, and et al. System entity structuring and model base management. pages 20: 1013–1024. IEEE Trans Syst Man Cybern, 1990.

[10] T. Ören and B. Zeigler. System theoretic foundations of modeling and simulation: a historic perspective and the legacy of a wayne wymore. pages 88: 1033–1046, 2012.

[11] B. Zeigler and P. Hammonds. Modeling and simulation-based data engineering: introducing pragmatics into ontologies for net-centric information exchange. London: Academic Press, 2007.

[12] H. Lee and B. Zeigler. System entity structure ontological data fusion process integrated with c2 systems. pages 4: 206–225. J Defense Model Simul Appl Methodol Technol, 2010.

[13] J. Rozenblit and B. Zeigler. Representing and construction of system specifications using the system entity structure concepts. In: Proceedings of the 1993 winter simulation conference, Los Angeles, CA, 12–15 Dec 1993.

[14] B. Zeigler. Object-oriented simulation with hierarchical, modular models: inteligent agents and endomorphic systems. San Diego, CA: Academic Press Professional, 1990.

[15] B. Zeigler. Multifaceted modeling and discrete event simulation. Orlando, FL: Academic Press, 1984.

[16] Bikash Chandra Karmokar, Umut Durak, Sven Hartmann, and Bernard P. Ziegler. Towards a standard computational representation for system entity structures.

[17] Umut Durak. Extending the knowledge discovery metamodel for architecture-driven simulation, modernization. page Vol. 91(12) 1052–1067. Simulation: Transactions of the Society for Modeling and Simulation International, 2015.

[18] B. P. Zeigler and P. E. Hammonds. Modeling and simulation-based data engineering: introducing pragmatics into ontologies for net-centric information exchange. Elsevier, 2007.

[19] H. S. Thompson, N. Mendelsohn, D. Beech, and M. Maloney. W3c xml schema definition language (xsd) 1.1 part 1: Structures. page W3C Working Draft Dec vol.3. The World Wide Web Consortium (W3C), 2009.

[20] Bernhard Kaiser. Application story of odd as part of safety assurance: The significance of a well-structured odd specification for the ad safety and sotif process. ANSYS, June 2021.

[21] Asam sim:guide, standardization for highly automated driving. ASAM e.V.

[22] U. Durak, S. Jafer, R. Wittman, S. Mittal, S. Hartmann, and B. P. Zeigler. Computational representation for a simulation scenario definition language. page page 1398. In 2018 AIAA Modeling and Simulation Technologies Conference.

[23] B. Chandra Karmokar, U. Durak, S. Jafer, B. N. Chhaya, and S. Hartmann. Tools for scenario development using system entity structures. page page 1712. In AIAA Scitech 2019 Forum.

[24] B. P. Zeigler and P. E. Hammonds. Modeling and simulation-based data engineering: Introducing pragmatics into ontologies for net-centric information exchange. Elsevier, 2007.

[25] Stefan Riedmaier, Thomas Ponn, Dieter Ludwig, Bernhard Schick, and Frank Diermeyer. Survey on scenario-based safety assessment of automated vehicles.

[26] Demin Nalic, Tomislav Mihalj, Maximilian Baeumler, and Matthias Lehmann. Scenario based testing of automated driving systems: A literature survey.

A Discrete Process Modelling Study of ARGESIM Benchmark 'C2 – Flexible Assembly System' with Warteschlangensimulator

Alexander Herzog

Simulation Science Center Clausthal-Göttingen / TU Clausthal, Arnold-Sommerfeld-Straße 6, 38678 Clausthal-Zellerfeld, Germany; *alexander.herzog@tu-clausthal.de*

SNE 34(1), 2024, 23-28, DOI: 10.11128/sne.34.bn02.10673
Received: 2024-02-10; Revised: 2024-02-19
Accepted: 2024-02-23
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. In ARGESIM benchmark "C2 – Flexible Assembly System" a multi station conveyor based production system is introduced. The aim is to analyze the model for bottlenecks and to maximize the throughput. This article studies the benchmark model using the discrete-event, stochastic simulation tool Warteschlangensimulator.

Introduction

This article describes and studies the solution of the ARGESIM comparison model "C2 – Flexible Assembly System" using the discrete-event, stochastic simulation tool Warteschlangensimulator. A full description of the considered model can be found in the SNE model definition [1].

The model consists of a conveyor system moving a fixed number of pallets in a circle. The pallets carry the workpieces and are loaded/unloaded at one station. The complete system consists of 8 stations which are connected to the conveyor belt. Each station can be targeted or bypassed by the pallets. Depending on the load of the stations a pallet may need multiple laps in the circle for full processing. The optimization goal is to choose the number of pallets circulating in the system in a way that maximizes the throughput of the system. Lead times of the workpieces and utilization of the different process stations are considered also. Based on the utilization of the stations improvement recommendations are made to further increase throughput.

1 Warteschlangensimulator

Warteschlangensimulator (see [2]) is a free and open source, platform independent, Java-based, event-driven, stochastic simulator. The permissive Apache 2.0 license allows to use the simulator in teaching, research and industrial/commercial context without restrictions.

The simulator allows graphical modelling of queueing systems in form of flow charts. Therefore over 100 station types are available. Inter-arrival times, service times etc. can be modelled using one of the 41 built-in probability distributions (including the option to map measured values as an empirical distribution). An automatic distribution fitter to find a distribution that matches measured values best is also available. For more complex definitions a formula parser is integrated; so for example shifted or truncated probability distribution can be used, too.

Models can be executed in animation mode showing the movement of the entities through the system including the display of queues and in a fast simulation mode without graphical output using multiple CPU cores for faster executing. Since the simulation runtimes are often in the range of a few seconds up to one minute, the effects of changes to input parameters can be investigated in a very interactive way.

During simulation all relevant statistic performance indicators are recorded automatically and are available via the built-in report viewer. Filtering and exporting the results is also possible. The fact that recording the performance indicators does not have to be configured manually in the model keeps the modelling of even large systems clear.

To handle complex control strategies, stations can optionally be extended using Javascript or Java code for branching, holding or changing entities passing through the stations.

While Javascript code is interpreted by an internal Javascript engine, Java code is compiled on the fly using the Java runtime environment running the simulator itself and then executed with full machine speed.

Warteschlangensimulator comes with English and German user interface, documentation and example models. A German text book about modelling and simulation using Warteschlangensimulator is also available, see [3].

The built-in parameter study function allows to easily evaluate a model for different parameter sets. This function was used to generate the results shown in figures 3, 4 and 5. If there is a clear target value for a statistic performance indicator and there are some input parameters which are identified as control values, the built-in optimizer can be used to automatically maximize or minimize the target value.

2 Model

A full description of the considered model can be found in the SNE model definition [1]. The conveyor belts are modelled in Warteschlangensimulator using delay stations. The delay times are calculated from the given belt speeds and the belt lengths. Since both values are fixed for each segment, the delay times are also fixed and deterministic. In general, delay stations have an unlimited capacity. The conveyor belt capacity restrictions (due to the belt lengths and the pallet sizes) are implemented using a decide station in front of the delay station which will only direct pallets to corresponding B2 processing area if there is enough space. The model consists of 8 subsystems as shown in Figure 1:

- Pallets arrive on the left side of the subsystem from the predecessor station. This arrival takes some time, which is modelled by the most left delay station.
- As the next step it is checked if there is enough free capacity on the upper processing line. This is done by the decide station. Also some heuristic considerations for shifting a pallet or to keep it on the B1 bypass main line are done here.
- If there is not enough space, the pallet will stay on the lower bypass lane and be moved to the exit via the main conveyor belt (B1 area).
- If there is free capacity in the B2 processing area, the pallet is shifted to the processing line (Sx).
- The upper processing line (B2) consists of two delay station on the left and on the right modelling the conveyor belt in the B2 area.
- The processing in the B2 area is modelled using a small delay station for the conveyor right at the process station and the process station itself. A process station can in contrast to a delay station only process a limited number of pallets at time. In this model the process stations can only handle one pallet per time.
- After processing the pallets are shifted back to the main conveyor belt (Sy).

The decision to shift to B2 or to stay at the B1 lane is made at each subsystem from the available capacity at B2, the need to process a pallet at a particular station and some heuristic considerations:

- There are three identically A2 stations (A2a, A2b and A2c). Each pallet only has to be processed at one of these stations. To balance the workload between the three A2 stations, the first will only take every third pallet which needs A2 processing, the second only every second pallet that needs A2 processing and the third accepts any pallet which needs to get A2 processing. (See appendix of this article for the exemplary Javascript code used to decide if an arriving pallet is to be processed at A2a.)
- Pallets needs processing either at the stations A3, A4 and A5 or at station A6 only. Station A3 is bypassed if there are less than three pallets in the B2 lane of A6 (since processing at A6 is faster than the sum of A3, A4 and A5 which would be needed as alternative).
- A4 is bypassed if A3 was bypassed (due to the heuristic rule or due to a too high load at A3) because in this case A6 (which covers the jobs done at A3, A4 and A5) has to be visited anyway. For the same reason A5 is bypassed if A3 or A4 processing was bypassed.

See Figure 2 for a schematic overview of the entire modelled system. At A1 the workpieces are loaded on the pallets and also the finished products are unloaded. Workpieces can be processed at one of the A2 stations (in the upper half of the main circle in the figure) first and then in the A3 to A6 area or vice versa.

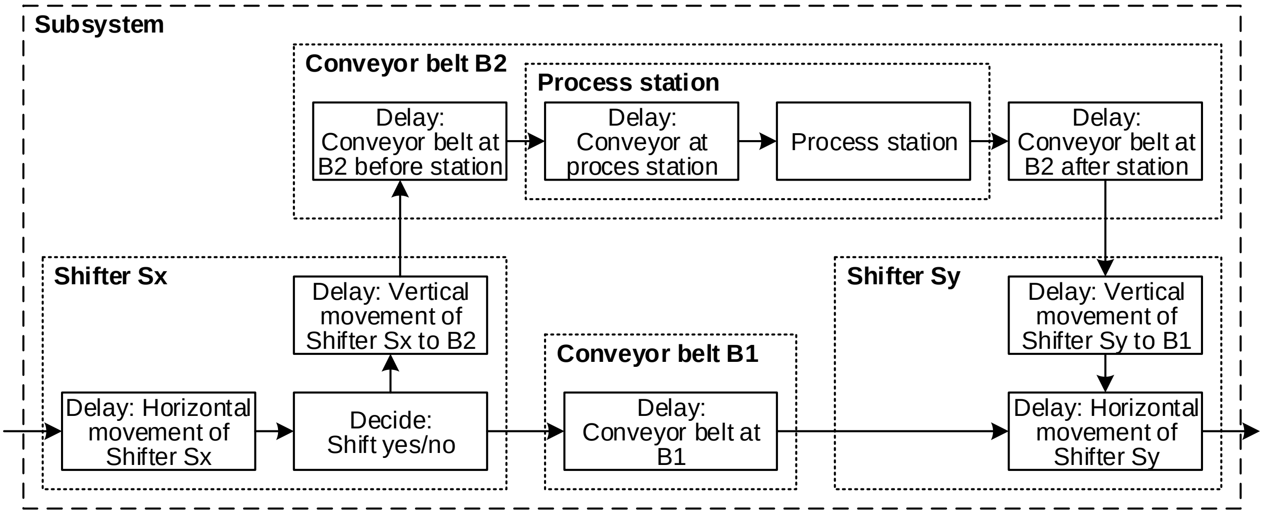


Figure 1: Schematic illustration of a single subsystem.

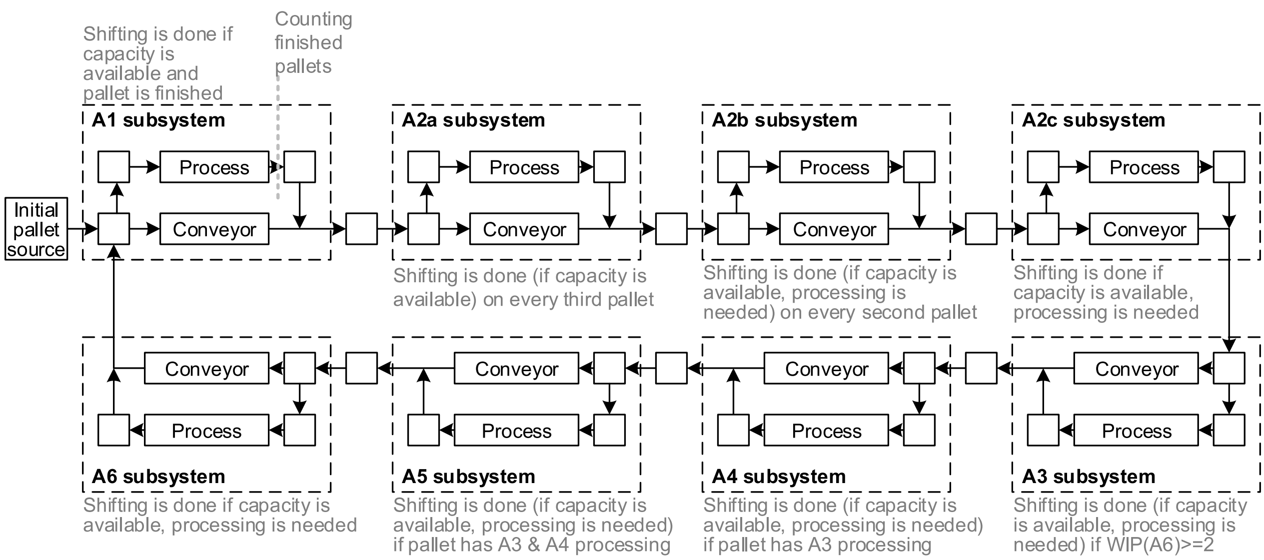


Figure 2: Schematic illustration of the entire system.

If the B2 processing areas of all possible stations in one area are blocked, a pallet may need to do several laps in the system, which will increase the lead time for the workpiece (the model files considered in this article are available for download, see URL in appendix).

3 Analytical considerations

- The minimum lead time of a pallet, which will be reached if there is only one pallet in the system, is 191.667 seconds:

Processing at one of the A2a to A2c stations (72 seconds) and bypassing the other two (8 seconds each), bypass stations A3, A4 and A5 (since A6 is empty; 8 seconds each), processing at A6 (43.333 seconds) and finally processing at A1 (28.333 seconds).
Additionally there are six short conveyor belts connecting the stations on the upper and the lower part of the model (1.333 seconds each; see the small boxes between the submodels in Figure 2).

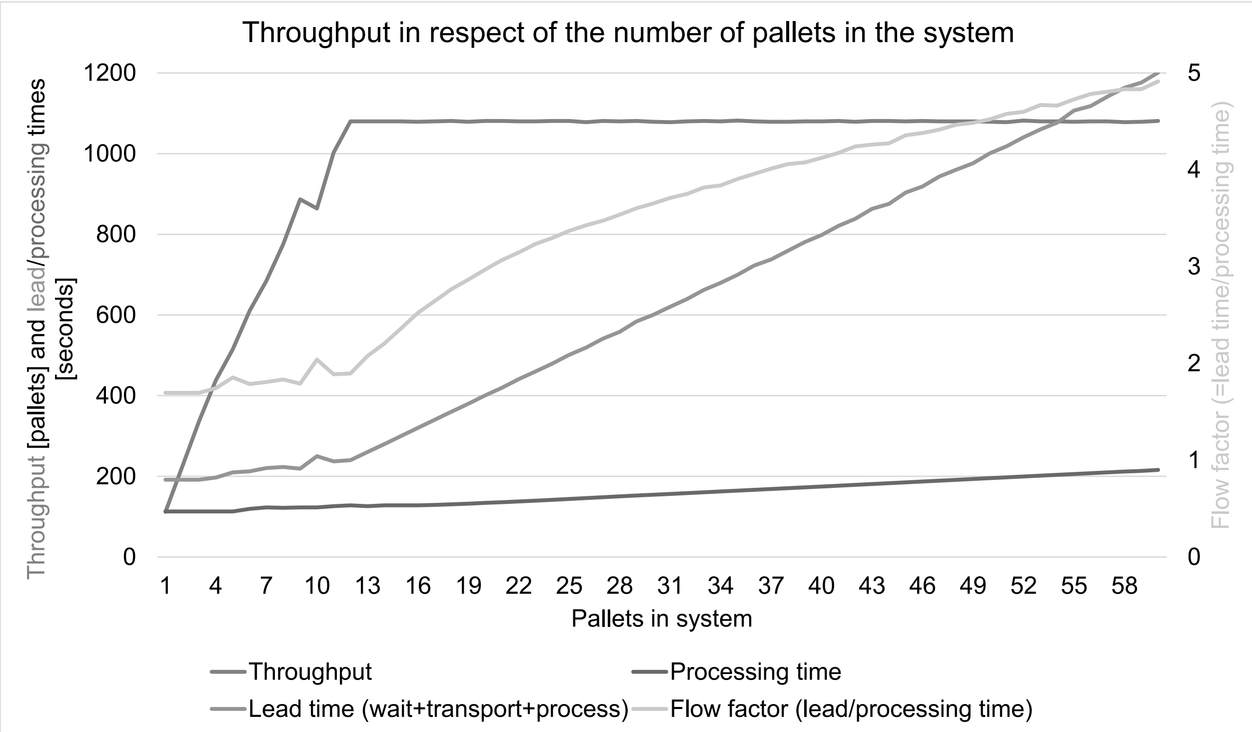


Figure 3: Throughput in respect of the number of pallets in the system.

In case of only one pallet in the system always the fastest processing options can be chosen and a workpiece will never need to do multiple laps in the systems resulting in the minimum possible lead time.
On the other side this would result in a throughput of only 112 workpieces within the considered time of 6 hours.
The simulation results (see red throughput graph on the left edge of Figure 3) confirm these considerations.

- From the simulation results (see blue graph in Figure 4) we will see that the A2 stations are the bottleneck of the system, i. e. these stations are running at 100 % utilization while there is still unused available work performance at the other stations.
The three A2 stations can handle one pallet per minute each. Therefore the maximum system throughput within 6 hours is 1080 pallets, which is confirmed by the throughput simulation results in Figure 3 (see red graph).

4 Simulation results

The key performance indicators in respect of the total number of pallets in the system are shown in Figures 3 (throughput, lead times, processing times and flow factor) and 4 (utilization of the individual stations). The total simulated time was 8 hours for each model, but the first two hours (used as warm-up phasis) are not recorded to statistics due to the benchmark definition.

Since the simulated model is a closed queueing network and there is a time valued used as termination condition for the simulation the automatic parallelization which Warteschlangensimulator offers on open queueing models can not be applied here and so the model is simulated using only one CPU core. Typical wall clock times needed to simulate the 8 hours are in the range below 0.5 seconds. The number of simulated events in this time is between 10,000 and 650,000 (depending on the number of pallets in system), i. e. over one million simulated events per second on models with higher number of pallets.

To get the maximum theoretically possible throughput of 1080 pallets within 6 hours, only 12 pallets in the system are needed (red graph in Figure 3). Increas-

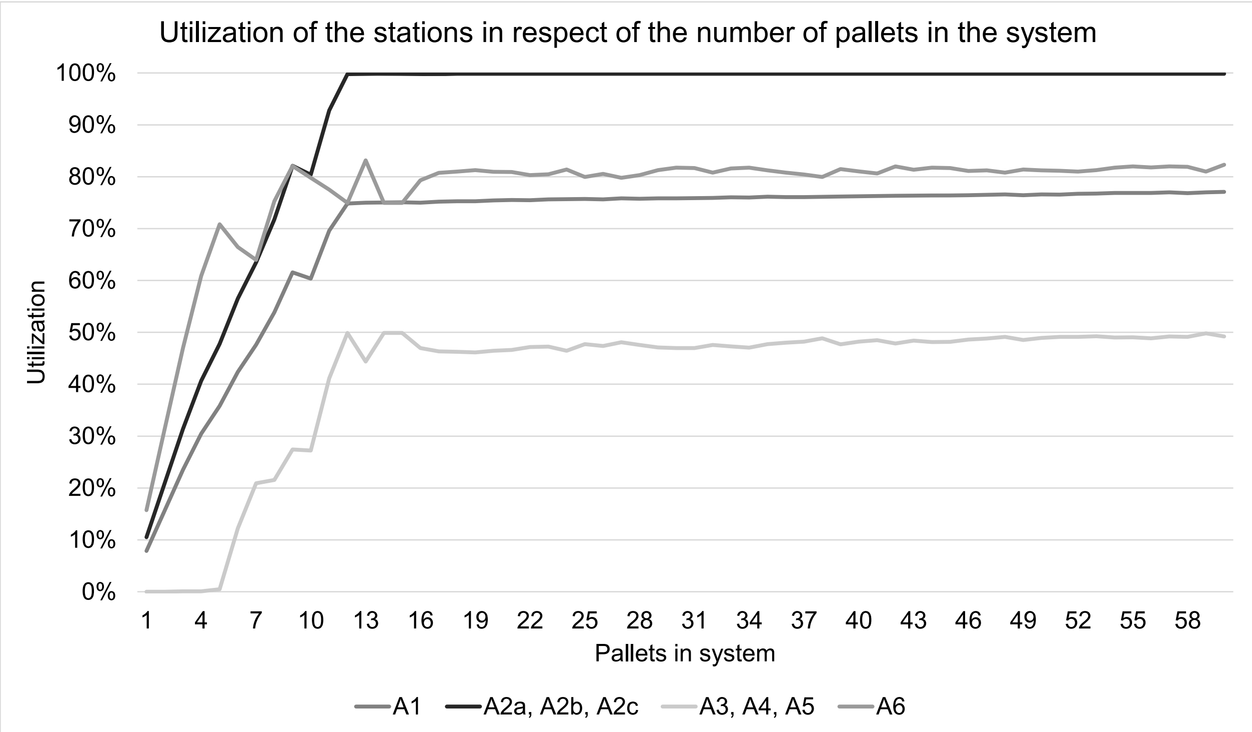


Figure 4: Utilization of the stations in respect of the number of pallets in the system.

ing the number of pallets further only increases the lead times (green graph) but not the throughput itself. The processing times (blue graph) are increasing also since on high load more pallets have to go through the longer A3, A4 and A5 path than the faster A6 only path.

5 Optimization

Figure 4 clearly shows that the A2 stations are the bottleneck which is limiting the throughput (blue graph). Decreasing the processing times at these stations from 60 to 50 seconds (i. e. by 16.6 %) would increase the throughput to 1295 pallets (by 19.9 % compared to the original model), see red graph in Figure 5. The new optimal number of pallets in the system would be 14.

Comparing Figure 3 and Figure 5 also shows that the average lead time and the flow factor is lower on identical number of pallets in the system. Since the A2 stations have been the bottleneck, increasing the performance at these stations reduces the number of pallets which have to do multiple laps in the system.

In the new model the A2 stations are still the bottleneck (and are working at 100 % utilization) but the utilization of the other stations is better (90 % at A1 and 87 % at A6 instead of 76 % and 82 % before). So increasing the available work performance at the A2 stations also leads to a better utilization of the available work performance at the other stations. When the available work performance at the A2 stations is increased in a way they are no longer the bottleneck, the next station to be considered for optimization would be A1.

References

[1] Model definition: C02 – Flexible Assembly System www.sne-journal.org/benchmarks/c02

[2] Warteschlangensimulator homepage a-herzog.github.io/Warteschlangensimulator

[3] Herzog, A. *Simulation mit dem Warteschlangensimulator*. Wiesbaden: Springer Gabler; 2021. 498 p.

Model files

The corresponding model files of the original model and the optimized model (shorter processing times at the A2 station) can be downloaded from:

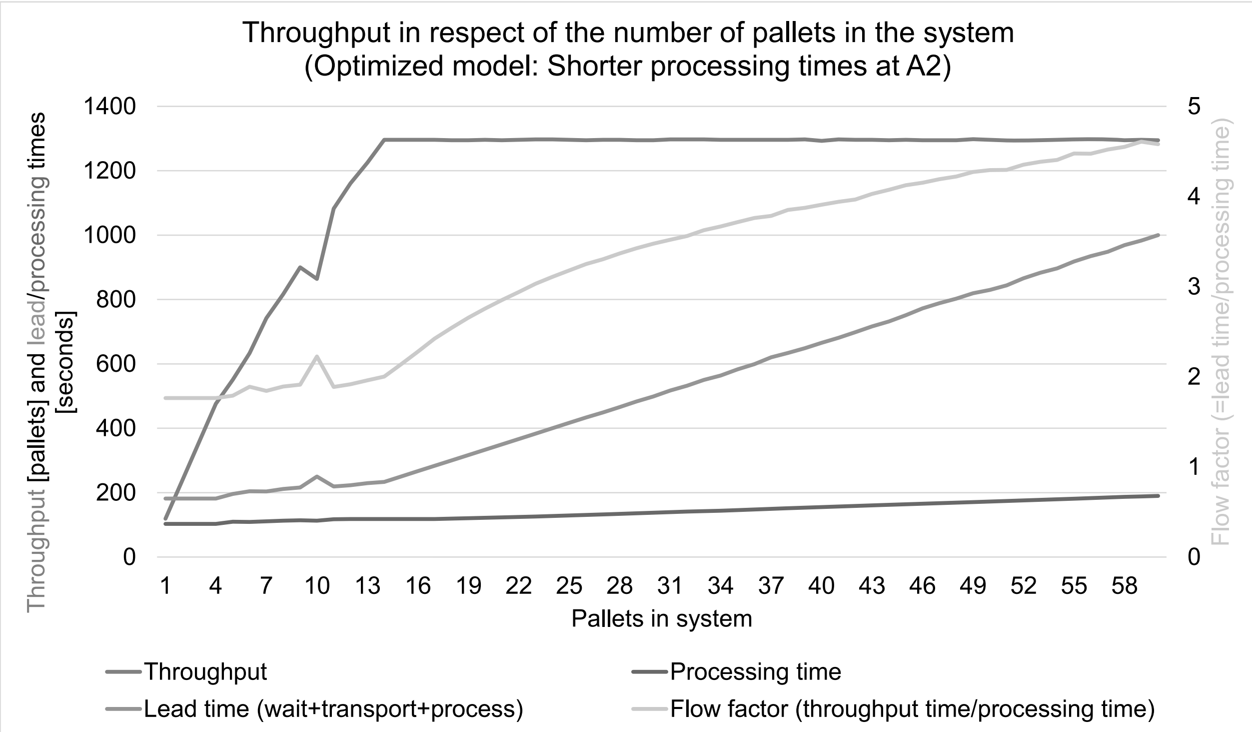


Figure 5: Throughput in respect of the number of pallets in the system – increased processing times at the A2 stations.

github.com/A-Herzog/Warteschlangensimulator
/tree/master/sne-benchmarks

The model `xml` files can be loaded directly into Warteschlangensimulator and simulated or animated there. The parameter studies (`zip` files) on which the Figures 3, 4 and 5 are created are also contained in the download package available via the URL above.

Javascript code for the shift decision at A2a

The following Javascript code is used for deciding if a pallet arriving at the A2a station is to be shifted to the B2 line for processing of if it will stay on the B1 lane and bypass processing.

```
// Get current number of pallets in B2 lane.
   let wipB2part1=Simulation.getWIP("A2a B2(1)");
   let wipB2part2=Simulation.getWIP("A2a horizontal");
   let wipB2part3=Simulation.getWIP("A2a");
   let wipB2=wipB2part1+wipB2part2+wipB2part;
// Count number of arriving pallets at this station
   let counter=Simulation.calc("A2aCounter")+1;
   Simulation.set("A2aCounter",counter);
// B2 has a capacity of 3. So processing there is
// only possible, if there are less than 3 pallets.
   let canB2=(wipB2<3);
// Does the pallet require processing at A2?
   let needA2=(Simulation.getClientValue(2)==0);
// Heuristic: Only process every third pallet
// passing A2a.
   let selectPallet=(counter%3==0);
// Decide if pallet is to be shifted.
   if (canB2 && needA2 && selectPallet) {
     // Status: A2 done
     Simulation.setClientValue(2,1);
     // Shift to B2
     Output.print(2);
   } else {
     // Stay on B1 lane
     Output.print(1);
   }
```

The code at the stations A2b and A2c is very similar. Only the `selectPallet` statement differs: A2b selects every second pallet passing the station and A2c every pallet which passes the station and needs processing.

A Discrete Process Modelling Study of ARGESIM Comparison ’C22 – Non-standard Queuing Policies’ with Warteschlangensimulator

Alexander Herzog

Simulation Science Center Clausthal-Göttingen / TU Clausthal, Arnold-Sommerfeld-Straße 6, 38678 Clausthal-Zellerfeld, Germany; *alexander.herzog@tu-clausthal.de*

SNE 34(1), 2024, 29-34, DOI: 10.11128/sne.34.bn22.10674
Received: 2024-02-10; Revised: 2024-02-19
Accepted: 2024-02-22
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. ARGESIM benchmark “C22 – Non-standard Queuing Policies” describes some queueing policies and customer behaviors beyond the default first-in-first-out rule. In this article it is studied how this policies can be implemented and investigated using the discrete-event, stochastic simulation tool Warteschlangensimulator.

Introduction

This article describes the implementation of the queueing policies and customer behaviors described in ARGESIM Comparison “C22 – Non-standard Queuing Policies” using the discrete-event, stochastic simulation tool Warteschlangensimulator. A full description of the considered policies can be found in the SNE model definition [1].

Thinking of a queueing system, first-in-first-out (FIFO) is usually assumed for the queue. But in reality frequently other concepts like choosing the shortest queue or individual priorities are used. Also impatience of the customers is a frequently occurring property not only of human customers but also in industrial context. This article shows how these concepts can be implemented and analyzed using Warteschlangensimulator.

1 Warteschlangensimulator

Warteschlangensimulator (see [2]) is a free and open source, platform independent, Java-based, event-driven, stochastic simulator.

The permissive Apache 2.0 license allows to use the simulator in teaching, research and industrial / commercial context without restrictions.

The simulator allows graphical modelling of queueing systems in form of flow charts. Therefore over 100 station types are available. Inter-arrival times, service times, waiting time tolerances etc. can be modelled using one of the 41 built-in probability distributions (including the option to map measured values as an empirical distribution). An automatic distribution fitter to find a distribution that matches measured values best is also available. For more complex definitions a formula parser is integrated; so for example shifted or truncated probability distribution can be used, too.

Models can be executed in animation mode showing the movement of the entities through the system including the display of queues and in a fast simulation mode without graphical output using multiple CPU cores for faster executing. Since the simulation runtimes are often in the range of a few seconds up to one minute, the effects of changes to input parameters can be investigated in a very interactive way.

During simulation all relevant statistic performance indicators are recorded automatically and are available via the built-in report viewer. Filtering and exporting the results is also possible. The fact that recording the performance indicators does not have to be configured manually in the model keeps the modelling of even large systems clear.

To handle complex control strategies, stations can optionally be extended using Javascript or Java code for branching, holding or changing entities passing through the stations. While Javascript code is interpreted by an internal Javascript engine, Java code is compiled on the fly using the Java runtime environment running the simulator itself and then executed with full machine speed.

The built-in parameter study function and a built-in optimizer allow to easily evaluate a model for different parameter sets and to optimize a model with regard to a target value.

Warteschlangensimulator comes with English and German user interface, documentation and example models. A German text book about modelling and simulation using Warteschlangensimulator is also available, see [3].

2 Models

A full description of the considered models can be found at [1]. All models considered in this article can be downloaded from:

github.com/A-Herzog/Warteschlangensimulator /tree/master/sne-benchmarks

In the model definition priority rules for concurrent events were defined. Since Warteschlangensimulator does not support a specific order for events to be executed at the same time, these priority rules will not be used in the following models.

2.1 Choosing the shortest line

A good heuristic for a customer to get the shortest waiting time in a system consisting of multiple, parallel stations is to choose the station where the fewest customers are located, see Figure 1.

The decide station at Warteschlangensimulator can directly select the following station with the shortest queue (NQ) or with the fewest entities at the station (WIP). Either the immediately following stations or the next following process stations on the path can be considered.

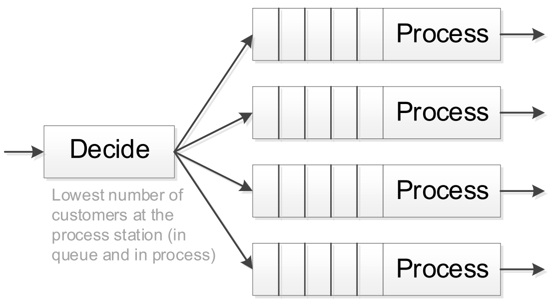


Figure 1: Schematic illustration of customers choosing the shortest queue.

The second option requires that there are no further decide stations on the path to the next process station, i. e. that the relevant process station for each path can be determined unambiguously. In case of tie between two or more stations, the station can be chosen randomly, top down or bottom up. For more complex decide rules (like in the jockeying example) also scripts (Javascript or Java code) can be used to branch the customers. A script code based decide station can also be used if the number of customers at the different stations is to be evaluated using different weights (i. e. customers on the "max. 5 items" express lane with a much lower weight then customers at the extra large shopping carts lane).

2.2 Jockeying

Jockeying means that the last customer in a queue in a system consisting of multiple parallel process stations with individual queues can change his queue if the queue on one of the other stations becomes shorter than the own queue. This behavior is usually combined with initially choosing the shortest queue.

The process station building block template available in Warteschlangensimulator consists of the actual process station and the queue in front of the station. To map jockeying of the last customers of the queue, the queue and the actual service process have to be split into two stations: Customers are served at a regular process station but the queue is organized at a script-based hold station before the process station. The script at these stations is triggered on each system state change (which covers the relevant events of a customer arrival and any state change at any process station). The scripts test two things:

1. If the process station ahead is empty, the first customer from the queue is released. The customer will then be sent to the process station for service.
2. If the jockeying condition is fulfilled, the last customer of the queue will get a "jockeying" flag and will be released. In this case the customer will be redirected to the decide station and will be sent to the shorter queue, see Figure 2.

2.3 Reneging Queues

Impatience of customers is a very common property in call center models and similar customer service systems. Human customers usually only have a limited tolerance to wait in a queue.

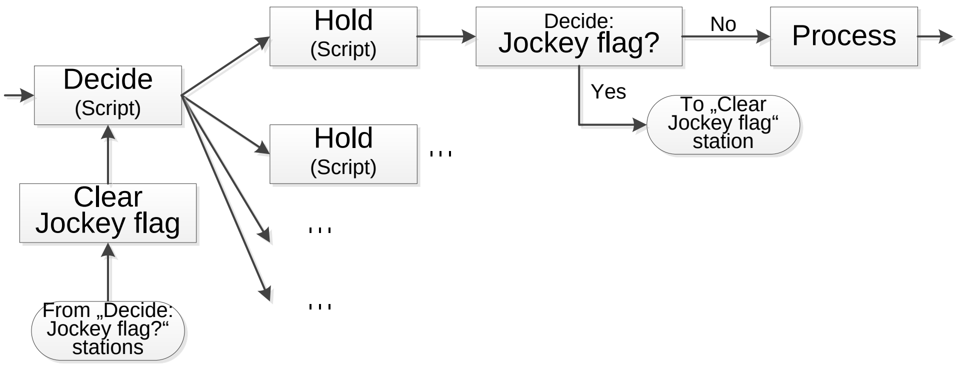


Figure 2: Schematic illustration of a waiting line with jockeying.

If this waiting time tolerance is exceeded, the customer will leave the system without being served. Impatience also occurs in industrial environments: If a workpiece can only be processed in a heated state, it cannot not wait too long after the oven process to be served at the next station.

If there are no special functions for mapping impatience directly in the simulator, handling impatience is quite hard: For each customer arriving at the queue a waiting time tolerance has to be calculated and stored with the customer object. At any time the waiting time tolerance property of each customer has to be checked and customers with an exceeded waiting time tolerance have to be removed on a special path not leading to the process station itself.

Warteschlangensimulator has a built-in impatience option for customers at process stations. This allows to add an impatience property for all customers or also for only some customers types arriving at the station with just one click. If a process station has two outgoing edges, one can be used for successful customers and the other for customers who have given up waiting. The waiting time tolerance (fixed value $t_R = 9$ in the model definition) can be any probability distribution (from which a random number is generated) or any calculation expression. Furthermore the waiting time tolerances formula of the customers can vary from customer type to customer type and can include customer specific data field values and runtime state information.

When simulating a model with impatience for each customer arriving at the process station queue an individual waiting time tolerance is calculated. The customer is added to the queue and a special "Waiting time tolerance exceeded" event is added to the events list in the simulation system.

A reference to this event is stored in the customer object. If the waiting time tolerance of the customer is exceeded, the event is executed and the customer is removed from the queue and forwarded via the "canceled customers" edge. If the customer is selected for service before his waiting time tolerance is exceeded, the corresponding event is searched in the events list and removed from there without being executed. This concept allows to handle impatience in a from the discrete-event point of view very light-weighted and therefore fast to be executed way.

2.4 Reneging Queues — Simulation results

When using stochastic waiting time tolerances for the customers, varying the average waiting time tolerance and measuring the waiting times of the customers who have canceled their waiting process (i. e. for which the individual waiting time tolerance was exceeded and therefore their waiting time tolerance equals their actual waiting time), one can see that the customers who have canceled the waiting process are usually the customers with the rather short waiting time tolerances.

Figure 3 shows the simulation results for a model with an average inter-arrival time of 100 seconds, an average service time of 80 seconds (both exponentially distributed) and one operator. The average waiting time tolerance has been varied from 20 to 500 seconds. The actual waiting time tolerances of the successful customers (red dashed graph) mirrors the global waiting time tolerance settings. But the customers who cancel the waiting process (red solid line) have shorter average waiting time tolerances and therefore are not representative for the entirety of the customers.

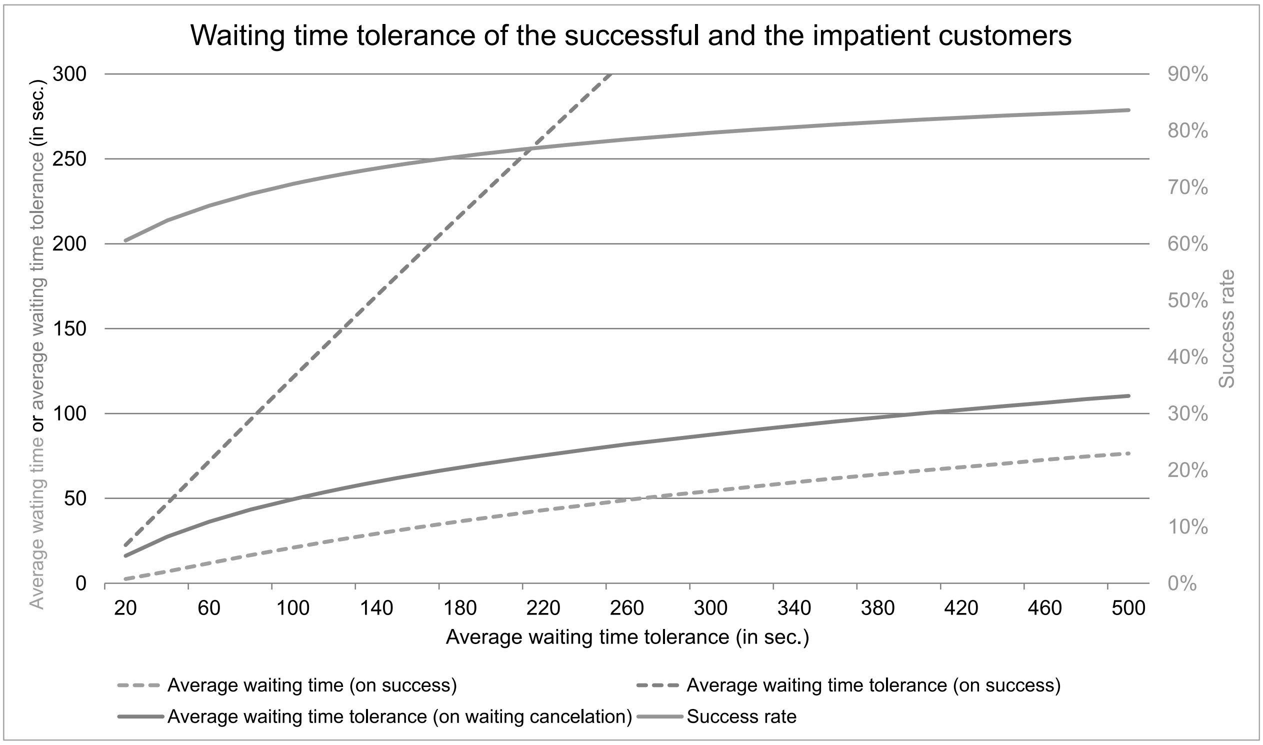


Figure 3: Waiting time tolerances of the successful and the impatient customers.

This is an interesting fact and a big problem when building simulation models for real customer service systems: For the simulation model a formula is needed to calculate the waiting time tolerances for all customers. But as input parameter only the waiting time tolerances of the customers who have canceled their waiting process (their actual waiting times) can be measured. The waiting time tolerances of the successful customers cannot be measured. For the successful customers it is only known that their waiting time tolerances obviously have been longer than their actual waiting times. So the not successful customers cannot be used as representative for all customers. This problem can be handled by using the method of parameter calibration. See [4] for more details about waiting time tolerance calibration in the context of modelling call center systems.

2.5 Classing Queues

In the model definition in [1], customers of a specific type are only to be served during their class is called. So there is no real resorting of the queue but instead there are $n_C = 5$ separate queues.

Depending on the currently active class only customers from one of these queues are sent to the process station. The concept of resorting the (single) queue is much more considered when also allowing the non-called classes to be served but to prioritize the called class (as shown in Figure 2 in [1]).

Since Warteschlangensimulator is not using a limited set of predefined queueing policies (like FIFO, LIFO etc.) but is always using priorities, this can be implemented very easily: Each customer object carries a user data vector which can be used for example to define the class of the current customer. Lets assume v_1 is set to 1 for a customer, if the customer is of class 1, and v_2 is set to 1, if the customer is of class 2 etc. Lets further assume the global variable $clsA$ will be set to 1, when class 1 is called etc. Then the service priority of a single waiting customer can be calculated as

$$w + 1000 \cdot v_1 \cdot clsA + 1000 \cdot v_2 \cdot clsB + \dots,$$

where w is the current waiting time at the station. If the class of the customer is not called ($v_1 \cdot clsA = v_2 \cdot clsB = \dots = 0$), the priority is the waiting time (which means FIFO policy between these customers).

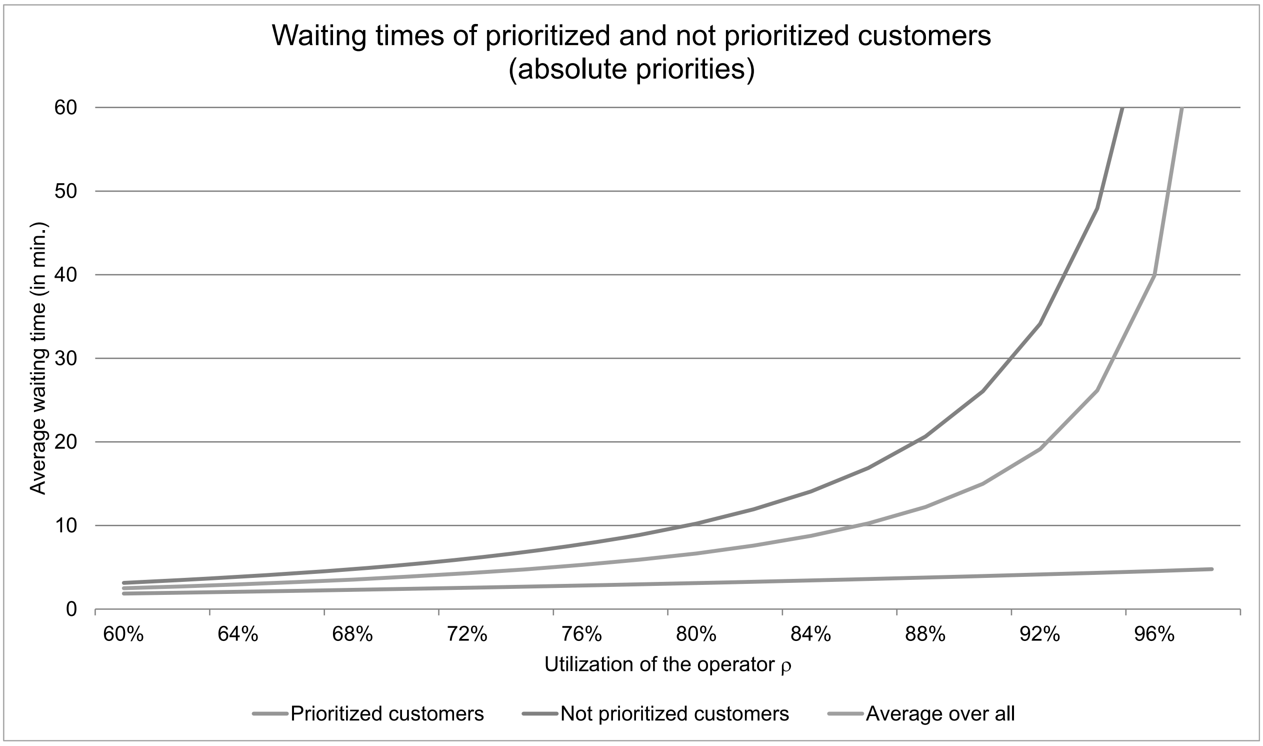


Figure 4: Waiting times of the prioritized and the not prioritized customers — using absolute priorities.

If class 1 is called ($clsA = 1$) and the customer is of this class ($v_1 = 1$) the priority is 1000 plus the waiting time (and the same for all other customers of class 1). This means the customers of class 1 will get 1000 seconds advantage in the queue during their class is called. Within all class 1 customers still FIFO is maintained. If all class 1 customers are served, the other customers will be served (also while maintaining FIFO).

2.6 Classing Queues — Simulation results

To show the effect of a strict prioritization a model with two customer types is considered: A prioritized class and a class of customers who is only served if there are no prioritized customers in the queue. Figure 4 shows the waiting times of the prioritized and the not prioritized customers when increasing the utilization of the system (and therefore increasing the average overall queue length). While the average waiting times of not prioritized customers increase quickly when increasing the utilization of the operator, the average waiting times of the prioritized customers increase only slightly.

So using different priorities allow to offer some customers a better service (shorter waiting times) in a system where customers of different types arrive. When using softer priorities (not only serving non prioritized customers if there are no prioritized customers at all but for example using different weights per waiting second in the priority formula), see Figure 5, the waiting time differences between the different customers types can be adjusted more finely.

References

[1] Model definition: C22 – Non-standard Queuing Policies
www.sne-journal.org/benchmarks/c22

[2] Warteschlangensimulator homepage
a-herzog.github.io/Warteschlangensimulator

[3] Herzog, A. *Simulation mit dem Warteschlangensimulator*. Wiesbaden: Springer Gabler; 2021. 498 p.

[4] Herzog, A. *Callcenter – Analyse und Management*. Wiesbaden: Springer Gabler; 2017. 437 p.

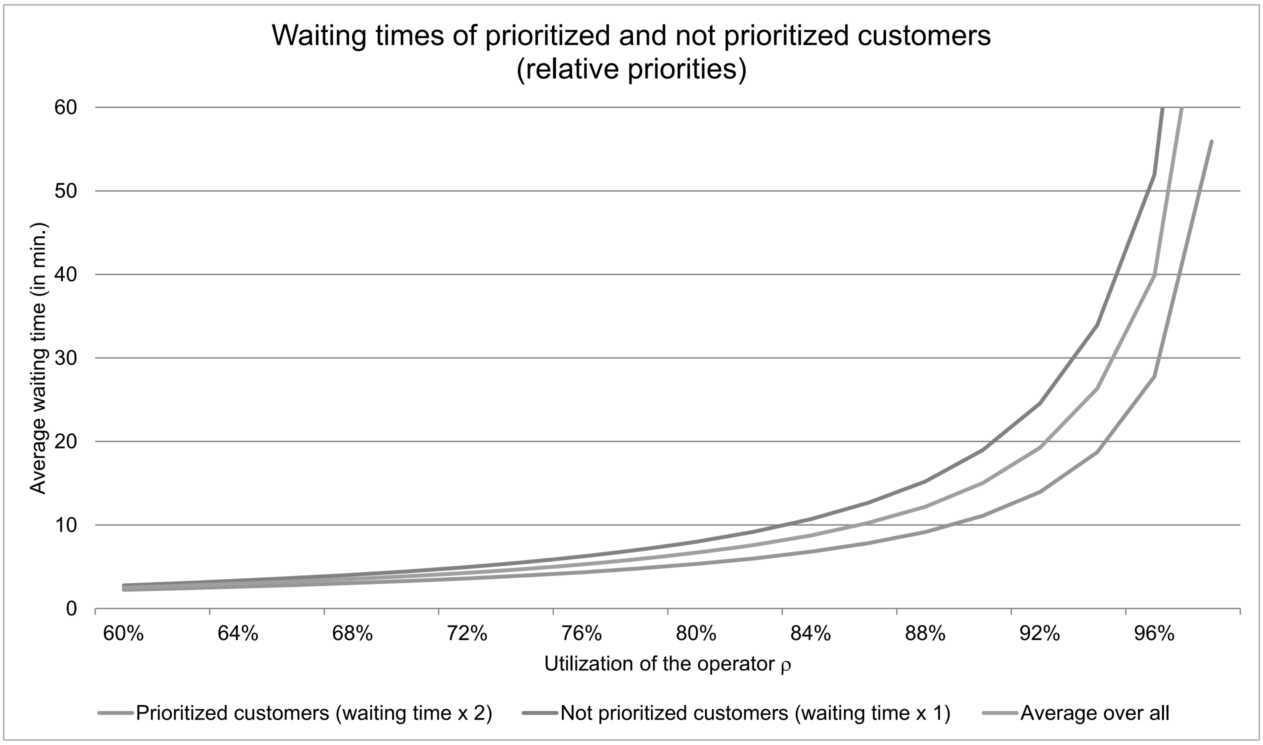


Figure 5: Waiting times of the prioritized and the not prioritized customers — using relative priorities.

Javascript code for mapping the jockeying property

The following Javascript code is used at the Javascript-based hold station in front of the first process station in the model shown in Figure 2.

```
// Which queue are we in?
   let ownRow="A";
   if (Simulation.getWIP(ownRow)==0) {
     // Server is free. Release first waiting customer.
     Clients.release(0);
   } else {
     // Customers at station A
     let wipAQueue=Simulation.getWIP("AQ");
     let wipAProcess=Simulation.getWIP("A");
     let wipA=wipAQueue+wipAProcess;
     // Customers at station B
     let wipBQueue=Simulation.getWIP("BQ");
     let wipBProcess=Simulation.getWIP("B");
     let wipB=wipBQueue+wipBProcess;
     // Customers at station C
     let wipCQueue=Simulation.getWIP("CQ");
     let wipCProcess=Simulation.getWIP("C");
     let wipC=wipCQueue+wipCProcess;
     // Customers at station D
     let wipDQueue=Simulation.getWIP("DQ");
     let wipDProcess=Simulation.getWIP("D");
     let wipD=wipDQueue+wipDProcessM
     // Minimum number of customers at a station
     let wipMin=Math.min(wipA,wipB,wipC,wipD);

     // Customers at the current station
     let wipOwnQueue=Simulation.getWIP(ownRow+"Q");
     let wipOwnProcess=Simulation.getWIP(ownRow);
     let wipOwn=wipOwnQueue+wipOwnProcess;

     // Is there a station with fewer customers?
     if (wipOwn>wipMin+1) {
       // Number of waiting customers
       let waitingCount=Clients.count();
       // Set jockeying flag for last waiting customer
       Clients.clientData(waitingCount-1,1,1);
       // Release last waiting customer.
       Clients.release(waitingCount-1);
     }
   }
```

Simulation-based Investigation of Energy Flexibility in the Optimization of Hinterland Drainage

Marvin Hempel[1*], Jens Heger[1]

[1]Institute for Production Technology and Systems (IPTS), Leuphana University of Lueneburg, Universitaetsallee 1, 21335 Lueneburg, Germany; *marvin.hempel@leuphana.de

SNE 34(1), 2024, 35-42, DOI: 10.11128/sne.34.tn.10675
Selected ASIM WS 2023 Postconf. Publication: 2023-10-15
Rec. Improved Engl. Version: 2024-02-28; Accepted: 2024-03-05
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. This study investigates the synchronization of pumping operations with dynamic energy prices in hinterland drainage systems. Employing mathematical models and optimization techniques, the research systematically evaluates the inherent flexibility within regular pumping operations. While traditional approaches often focus solely on tidal dynamics to enhance energy efficiency, this study introduces a more comprehensive perspective. Beyond the conventional scope, it incorporates a nuanced analysis of electricity price patterns, aiming to promote grid-friendly behavior and foster an adaptive and efficient response to fluctuating energy availabilities. By delving into these complex interplays, the research not only deepens the understanding of hinterland drainage system dynamics but also proposes an approach for robust and sustainable operational practices. Moreover, the holistic approach aligns with the broader objective of reducing CO2 emissions, reflecting a commitment to environmentally responsible infrastructure management. Through this balanced exploration of operational optimization potentials, the research contributes to the scientific discourse on energy-efficient and environmentally conscious practices in hinterland drainage systems.

Introduction

The ongoing energy turnaround in Germany is changing the way we approach energy-intensive facilities. With the increasing share of renewable energy sources in the power grid, the fluctuation in energy availability naturally rises [1].

These fluctuations directly influence energy prices in the electricity market. Generally, electricity prices tend to be lower when there is a high contribution from renewable sources, and vice versa. In addition to the concept of energy efficiency, there is an additional potential for flexibility. By utilizing this flexibility, operators can reduce energy costs, simultaneously lower indirect CO2 emissions, and relieve the strain on the power grid.

As a result, the cost- and energy-intensive pumping processes in hinterland drainage should be adapted to the significantly more complex conditions in today's electricity market through intelligent control systems. This adaptation aims to ensure holistically optimized drainage strategies. The intensification of rainfall events due to climate change, coupled with the increasing electricity prices, poses challenges for pumping station operators aiming to implement appropriate drainage measures distinct from outdated concepts. In certain cases, relying solely on manual binary decisions and operating pumps at full capacity during high internal or low external water levels may prove insufficient.

By implementing pump speed control, there is the possibility to achieve different efficiencies, with lower speed resulting in significantly reduced energy consumption. However, a longer period is necessary to discharge the same volume of water, and the maximum discharge head that can be overcome is reduced. The choice of the optimal speed depends directly on the current water levels, their rate of change, and the current electricity price at any given time.

To counteract the global increase in CO2 emissions and rising energy costs, extensive knowledge must be generated to understand the complex relationships between controllable factors, such as pump duration and pump speed, and uncontrollable factors like tides, water levels, and weather conditions in a drainage system.

Consequently, a flexible and supply-oriented consumption in energy can be achieved by utilizing forecast data for precipitation, tides, and power availability in the future.

1 State of the Art

The field of energy flexibility is a highly topical issue due to the prevailing energy turnaround, and it is being researched across various industries [2]. In the work by Reinhart et al., energy flexibility is described as the ability of production systems to adapt rapidly and cost-effectively to changes in the energy market [3].

In literature, the inclusion of energy consumption data into classical simulation models for production process planning enables energetic optimizations. By leveraging the temporal flexibility of individual production steps, an optimized adaptation to fluctuations in energy supply or electricity prices can be implemented [4, 5].

Similarly, for applications within the cost- and energy-intesive water management sector, possibilities are being explored to increase energy efficiency of process steps and reduce costs. Mathematical models are typically formulated to represent the mathematical relationship between the in- and outflow of water in a tank or tank cascade. The literature predominantly focuses on water supply or urban drainage [6, 7], with little attention given to drainage systems in coastal regions. However, especially for hinterland drainage, novel methods for drainage optimization arise due to fluctuations in the electricity market. Traditional methods of pumping during low tide are now being contrasted with opportunities for more complex planning of pumping processes that consider the fluctuating electricity price structures [8, 9].

2 Scenario Description

The Kehdingen Maintenance Association (Unterhaltungsverband Kehdingen in German) is one of the 114 associations in Lower Saxony, Germany. Encompassing an area of approximately 27,000 hectares along the Elbe River and its immediate environs, the association is engaged in the operation and management of a comprehensive network comprising approximately 160 pumping stations, housing 400 pumps. The central mandate of this maintenance association revolves around the strategic diversion of water originating from the natural precipitation area.

Given the geographical characteristics of the region, characterized by its low altitude (below sea level) natural drainage becomes a rare occurrence. In instances where natural drainage is unattainable, the accumulated water within the maintenance area necessitates deployment of a considerable energy-intensive mechanism, involving the pumping of water into the surrounding tidal waters. In the past, this process has incurred a substantial annual energy consumption from approximately 2.5 to 3 million kWh, for the maintenance association and its members. Consequently, this energy consumption in the year 2021 corresponded to an average emission output of approximately 1300 tons of CO2 equivalents.

In the context of a conventional drainage system, a diverse array of pumping stations assumes distinct roles, each characterized by specific functions that depend upon its position within the system. Polder pumping stations, for instance, are instrumental in collecting water designated for discharge directly inland, directing it through pump and pipe towards elevated storage basins or canals. Positioned at higher elevations, these storage basins offer two alternative pathways for the water: it can either undergo further elevation through second-stage pumping stations to attain an even greater height, or it can be promptly channeled into neighboring tidal waters by dike pumping stations. The quantity of distinct pumping stations exhibits variability and is tailored to suit the conditions inherent in each locale. Determining factors in this configuration encompass considerations such as the geographical expanse of the area and the overall lift requirement. For instance, there exists a relatively abundant distribution of polder pumping stations situated inland, with second-stage pumping stations strategically deployed on an as-needed basis. In contrast, a comparably limited number of dike pumping stations are strategically positioned at the edges of dikes. In terms of capacity, dike pumping stations inherently have significantly higher capabilities than their counterparts, facilitating the efficient processing of all precipitation water to be discharged from the enclosed area.

Certain pumping stations within the drainage system are equipped with fixed overflow edges, signifying that the water designated for discharge is consistently pumped to an elevated level above the water surface of the subsequent storage basin via an integrated pipe system. The drained water then flows into the next collecting basin. Importantly, the lift height at any given moment is contingent solely upon the prevailing water level within the upstream collecting basin.

Particularly in dike pumping stations, a distinctive operational concept is employed, involving the pumping of water beneath the dike into tidal waters. Notably, the water level within the tidal waters typically exceeds the discharge point of the pump, as illustrated in Figure 1. Consequently, within this concept, the prevailing lift height is dictated by the disparity between the external water level in the tidal waters and the internal water level within the upstream sections.

Given the cyclic variations in tidal water levels, attributed to tidal fluctuations, intervals characterized by relatively high or low lift heights emerge periodically. These fluctuations introduce dynamic variations in the lift heights, accentuating the inherent dynamism associated with the dike pumping station's operational paradigm.

To conduct a comprehensive analysis of specific attributes of the system amid the variability of water levels, a systematic collection of data is undertaken at diverse parameter states crucial to the system's functionality. This process entails the adjustment of the motor speed of the pump, employing a discrete sinusoidal pattern within the operational range under authentic conditions.

This pattern spans the entirety of the operational spectrum, ranging from 65% to 100% of the nominal rotational speed specific to the retailer. The data collection process is facilitated through the implementation of specific measurement systems. These encompass recording of flow rate, power consumption, and water level measurements.

The utilization of these measurement systems ensures capturing an extensive array of system-related data points, contributing to a comprehensive understanding of the system's performance characteristics.

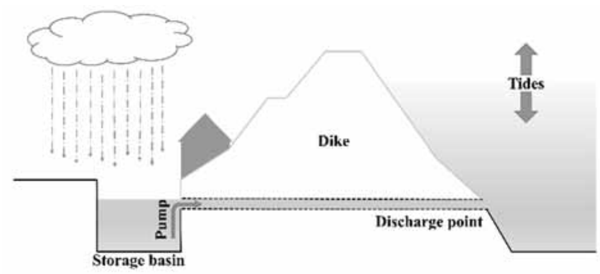


Figure 1: Schematic diagram of a pumping station for pumping rainwater under the dike into tidal waters.

3 Mathematical Model

This section presents the mathematical modeling of a single dike pumping station with a variable-speed pump. Based on this model, different scenarios are compared in terms of a mathematically optimized drainage strategy.

3.1 Model Description

The basis for the subsequent mathematical modeling is the Mixed-Integer Nonlinear Optimization (MINLP) framework presented in the work by Fecarotta et al. [6]. Given that drainage systems typically consist of a network of multi-stage pumps and storage basins exhibiting systematic interdependence, the problem is transformed into a Mixed-Integer Linear Optimization Problem (MILP) using linearization and discretization techniques. This reduction in complexity aims to ensure the calculation of multi-stage pumping schedules within reasonable time constraints under real-world conditions.

Initially, the classification of measured, system-specific data points for the pump results in $N = 7$ equidistant speed ranges (N_1: 65%–70%, N_2: 70%–75%, …). Subsequently, the application of linear regression methods allows the derivation of discretized, linear system characteristic curves.

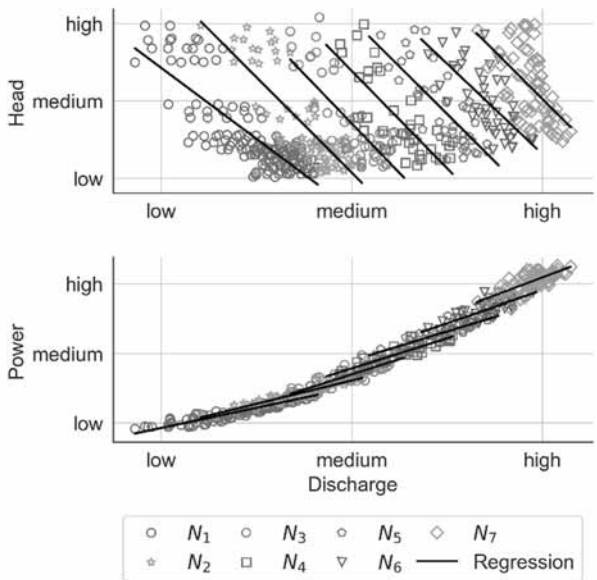


Figure 2: Measured data points with resulting system characteristics of an installed drainage pump in the speed range (N_1 - N_7) from 65 % to 100 % of the specific nominal speed.

These curves define the relationship between head or power and flow rate of the pump, providing insights into the general functioning of a corresponding pump in its operational range (see Figure 2).

Four different vectors are generated to represent the characteristic curves. Vectors M_{H} and M_{P} capture the slope of the regression lines for the head and power characteristic curves, respectively. B_{H} and B_{P} represent the ordinate interception of the curves.

The mathematical optimization in this work is based on a multicriteria minimization of the sum of the accumulated energy consumption P_{t} and energy costs K_{t} with a weighting factor w:

$$\min ZF = \sum_{t \in T} (w \cdot P_{\mathrm{t}} + \frac{(1-w)}{s} \cdot K_{\mathrm{t}}) \cdot dt \tag{1}$$

Here, T represents the number of periods, and dt is the duration of the planning periods. Energy costs are derived from the energy consumption and the energy prices p_t of individual periods ($K_t = P_t \cdot p_t$). A scaling factor s adjusts the costs to the range of energy consumption values, with s representing the average value of p_t. The preference for different objective criteria can be varied arbitrarily through the weighting factor w.

The decision variables of the optimization model are represented by binary variables $x_{t,n} \in \{0,1\}$, where $n \in \{0,1,\dots,N\}$ and $x_{t,0} = 1$ signifies the deactivation of the pump in period t. For all $n \in \{1,\dots,N\}$, $x_{t,n}$ represents the choice of one of the N specific speed ranges.

The system is subject to various constraints, which limit the optimization possibilities by restricting the solution space. These constraints are formally presented in the following.

Through a discretized continuity equation

$$H_{\mathrm{in}_t} = H_{\mathrm{in}_{t-1}} + \frac{dQ_{\mathrm{in}_t} - dQ_t}{S} \cdot dt \tag{2}$$

it is ensured that the change in the internal water level H_{in_t} in the reservoir is consistent with the average discharge

$$(dQ_t = \frac{Q_t + Q_{t-1}}{2})$$

and inflow

$$(dQ_{\mathrm{in}_t} = \frac{Q_{\mathrm{in}_t} + Q_{\mathrm{in}_{t-1}}}{2})$$

of water in consecutive time intervals.

These values are based on the discharge Q and inflow Q_{in} between two consecutive time intervals.

As the water level of the tank should not fall below a certain lower (c_{lb}) or exceed the upper (c_{ub}) bound, capacity constraints are defined, which represent an idle and overflow protection:

$$c_{\mathrm{lb}} \leq H_{\mathrm{in}_t} \leq c_{\mathrm{ub}} \tag{3}$$

The head H_t in each time interval is defined as a function of the internal H_{in_t} and external H_{ext_t} water levels:

$$H_t = H_{\mathrm{ext}_t} - H_{\mathrm{in}_t} \tag{4}$$

The following four inequalities allow discrete selection from the N available regression curves for different speed ranges.

By choosing sufficiently high values for $H_{\max}$ and $P_{\max}$ the constraints for non-selected speed ranges $x_{t,n}\ \forall n \in N \backslash \{0\}$ are relaxed to avoid limiting the available solution space.

$$M_{H_n} \cdot Q_t + B_{H_n} - H_{\max} \cdot (1 - x_{t,n}) \leq H_t \tag{5}$$

$$M_{H_n} \cdot Q_t + B_{H_n} + H_{\max} \cdot (1 - x_{t,n}) \geq H_t \tag{6}$$

$$M_{P_n} \cdot Q_t + B_{P_n} - P_{\max} \cdot (1 - x_{t,n}) \leq P_t \tag{7}$$

$$M_{P_n} \cdot Q_t + B_{P_n} + P_{\max} \cdot (1 - x_{t,n}) \geq P_t \tag{8}$$

The following two formal relationships ensure that the generatable flow or required power is set to zero if the pump is deactivated. By choosing sufficiently high values for $Q_{\max}$ and $P_{\max}$ the constraints for an active pump are relaxed to avoid restricting its operation in the working area.

$$(1 - x_{t,0}) \cdot Q_{\max} \geq Q_t \tag{9}$$

$$(1 - x_{t,0}) \cdot P_{\max} \geq P_t \tag{10}$$

Additionally, it is defined by the following equation that the pump is either deactivated within a time interval ($n = 0$) or exactly one speed range of the pump is activated ($n \in N \backslash \{0\}$).

$$\sum_{n=0}^{N} x_{t,n} = 1 \tag{21}$$

For reasons of plausibility, non-negativity conditions apply to both flow and power:

$$Q_t \geq 0 \quad (32)$$

$$P_t \geq 0 \quad (43)$$

3.2 Experiment Execution

During the execution of the subsequent experiments, various necessary supplementary data values are incorporated into the presented MILP. The Gurobi Solver is utilized to compute an optimal pumping schedule for four consecutive days (or 384 time steps with a time window of 15 minutes).

In addition to implementing three different electricity price patterns (Price 1, 2, and 3) exhibiting varying degrees of electricity price fluctuations (refer to Figure 3), the tidal behavior of a tidal water body is simulated using a sinusoidal function with a 12-hour cycle.

The initial water level of the collecting basin is set to a fixed value, positioned at half the height of the water level suitable for discharge. Furthermore, a linear storage cascade is employed to convert precipitation into runoff data, modeling the inflow to the collecting basin [10]. In this context, an inflow pattern consisting of three randomly consecutive inflows with varying intensity is selected. Additionally, the inflow pattern is scaled by a factor of 1.5 to model a second, stronger inflow, allowing for the examination of different effects.

In Figure 4, an exemplary representation of a single scenario with optimized pump planning over a time horizon of four days is depicted. The first row of the diagram illustrates the underlying electricity price patterns.

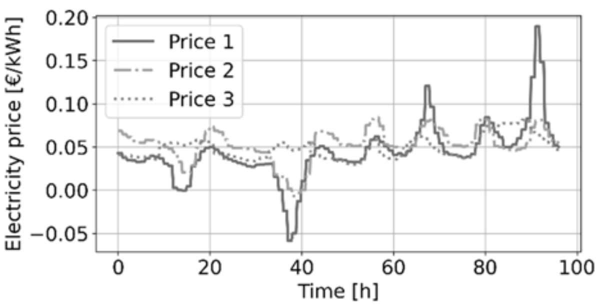


Figure 3: Four-day trends with quarter-hourly resolution of different past wholesale prices for electricity in Germany/Luxembourg based on data from the Federal Network Agency [11].

The second row visualizes the tidal-like behavior of the external water level. Following this is the resulting internal water level of the collecting basin through optimization, starting at the mean of the capacity limits (lower bound: 0.6 m, upper bound: 0.8 m). Row 4 displays the applied inflow pattern, and in the subsequent three rows, the optimal profiles of pump performance data (flow rate, speed, and power) are shown.

Based on the specified conditions, six different study scenarios arise, as presented in Table 1. Each scenario utilizes one of the three electricity price patterns and one of the two inflow intensities.

Within these scenarios, the weighting factor w is varied in equidistant steps between 0 and 1, allowing for the exclusive consideration and optimization of either energy costs or energy consumption, or a weighted combination of both factors.

In order to ensure comparability between the results using different price patterns, a simulation is conducted where the mathematical optimization is applied to 38 cyclic shifts of the inflow pattern.

In each optimization run, the data series is shifted by ten time steps. This shift ensures that different random parameter states occur between current price and inflow, introducing variability.

	Weak inflow	Strong inflow
Price 1	Relatively strong price fluctuations with moderate precipitation	Relatively strong price fluctuations with heavier precipitation
Price 2	Average price fluctuations with moderate precipitation	Average price fluctuations with heavier precipitation
Price 3	Relatively weak price fluctuations with moderate precipitation	Relatively weak price fluctuations with heavier precipitation

Table 1: Presentation of the study scenarios for different electricity price patterns and inflow intesities.

The evaluated metrics include the resulting average energy consumption and average energy costs. The basis for the six scenarios in Table 1 is determined by the optimization results obtained, assuming a constant average price for the underlying electricity price pattern. This assumption serves as a representative benchmark for the respective scenario and is also representative for the optimization of tariff-based electricity procurement.

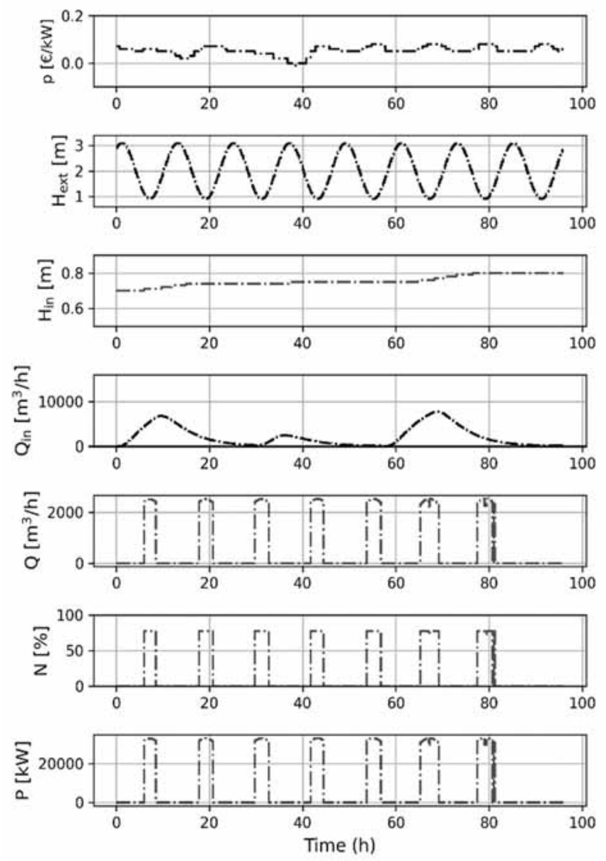


Figure 4: Exemplary optimized pump scenario for four days with fluctuating electricity prices (Price 2) and weak inflow pattern for $w = 1$.

4 Results

The findings presented in Figure 5 illustrate the alteration in the pump pattern corresponding to variations in the weighting factor.

The foundational approach, prioritizing the minimization of energy consumption, defined as $w = 1$, manifests in the planning of requisite pumping operations primarily during phases of relatively low tide. In contrast, it is equally apparent that the strategy of minimizing energy costs with $w = 0$ prompts an adaptation of pumping operations to coincide with periods characterized by lower energy expenses.

Utilizing the data presented in Figure 6, a more in-depth analysis uncovers that, across all scenarios listed in Table 1, the escalation of values for the variable w leads, on average, to a reduction in energy consumption and an increase in energy costs.

Notably, the costs demonstrate a relatively robust exponential growth in this context. Observing from the upper to lower perspective, this effect experiences a swift reduction with decreasing volatility in the electricity price pattern. Similarly, when examining from left to right, the augmentation in inflow intensity mitigates this effect as well.

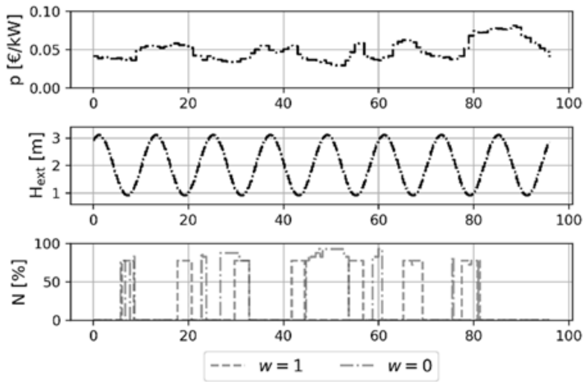


Figure 5: Comparison for the optimization of energy consumption and costs based on exemplarily optimized pump scenarios for four days with a random inflow pattern (Price 3).

Utilizing the data presented in Figure 6, a more in-depth analysis uncovers that, across all scenarios listed in Table 1, the escalation of values for the variable w leads, on average, to a reduction in energy consumption and an increase in energy costs.

Notably, the costs demonstrate a relatively robust exponential growth in this context. Observing from the upper to lower perspective, this effect experiences a swift reduction with decreasing volatility in the electricity price pattern. Similarly, when examining from left to right, the augmentation in inflow intensity mitigates this effect as well.

When using constant average prices, $w > 0$ leads to constant average values for energy consumption or energy costs, as the cost optimization criterion offers no advantages due to the absence of electricity price fluctuations. Thus, the resulting energy consumption always corresponds to the consumption for $w = 1$.

However, a different pattern emerges for energy costs. In the presented range, costs at constant electricity price structures exhibit a high negative discrepancy for $w < 1$, as no low-price phases can be utilized for pumping operations.

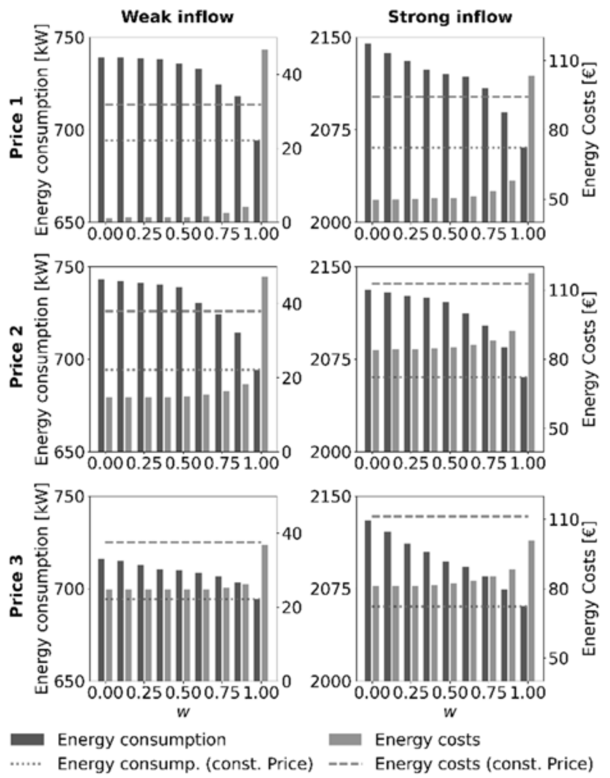


Figure 6: Simulation results for the mathematical optimization of the objective criteria (energy consumption and energy costs) under variation of the weighting factor w for different inflow scenarios and electricity price patterns.

5 Summary

In the context of this research contribution, a simplified Mathematical Model (MILP) was presented, designed to elucidate critical relationships within a drainage system. This model employs complex reduction techniques such as linearization and discretization. The MILP was strategically employed to conduct a comparative analysis of various scenarios, specifically focusing on energy consumption and associated costs.

The study showcased the extent to which costs can be mitigated under specific conditions by integrating considerations of electricity price fluctuations into the strategic planning of pumping operations.

Through extensive simulation efforts, the obtained results were rigorously tested across a diverse range of external conditions.

Notably, our findings underscore the substantial cost-reducing effects achieved through even a modest prioritization of energy costs in the optimization process.

To advance the scope of this research, a promising avenue is the in-depth analysis of multi-stage drainage systems. Furthermore, the exploration of stochastic fluctuations in electricity prices, precipitation, or inflows could yield additional valuable insights.

For example, such an approach could delve into deviations of solution quality from the computed optimum, providing a more comprehensive understanding of the system's dynamic behavior.

Acknowledgements

This research project is deeply grateful for the financial support received through the collaborative initiative "SCHOEPFWERK40 – Reduktion des Energieverbrauchs von Schöpfwerken durch intelligente Pumpsteuerung und -regelung," specifically under grant number 03EN2008 A-C. This funding was provided within the framework of the 7th Energy Research Program by the Federal Ministry of Economics and Climate Protection (BMWK, Germany).

Furthermore, our sincere appreciation is extended to the Lower Saxony Ministry of Science and Culture (MWK, Germany) for their generous support of the “Zukunftslabor Wasser,” recognized by grant number ZN3994 and managed by the Center for Digital Innovation Lower Saxony (ZDIN, Germany). It is important to highlight that this support played a crucial role in facilitating the publication of this paper, enabling the dissemination of our research findings.

The supplementary research endeavors carried out as part of this project were facilitated by the subproject “In-Was – Indikatorbasierte Wasser- und Energiestrategie zur Be- und Entwässer-ung.”

We acknowledge the collaborative efforts and commitment of our esteemed project partners, Mr. Nicolai, Managing Director and Technical Management of Unterhaltungsverband Kehdingen, and Mr. Flathmann, Head of Technical Office at Vetterkind GmbH. Their invaluable support has significantly contributed to the success and progress of the project throughout its duration.

Editors Note. This contribution is an improved and into English language translated version of a contribution to ASIM Workshop 2023, Magdburg, Germany, in the Proceedings published as:

Hempel M, Heger J. *Simulationsbasierte Untersuchung von Energieflexibilität bei der Optimierung in der Hinterland-entwässerung*. DOI: 10.11128/arep.21.a2109
In ARGESIM Report 21, Proceedings Langbeiträge ASIM Workshop STS-GMMS-EDU 2023, Magdeburg, März 2023, ISBN 978-3-903347-61-8.

References

[1] Bachmann A, Bank L, Bark C, Bauer D, Blöchl B, Brugger M, Buhl HU, Dietz B, Donnelly J, Friedl T, Halbrügge S, Hauck H, Heil J, Hieronymus A, Hinck T, Ilieva-König S, Johnzén C, Koch C, Köberlein J, Köse E, Lochner S, Lindner M, Mayer T, Mitsos A, Roth S, Sauer A, Scheil C, Schilp J, Schimmelpfennig J, Schulz J, Schulze J, Sossenheimer J, Strobel N, Tristan A, Vernim S, Wagner J, Wagon F, Weibelzahl M, Weigold M, Weissflog J, Wenninger S, Wöhl M, Zacharias J, Zäh MF. *Energieflexibel in die Zukunft - Wie Fabriken zum Gelingen der Energiewende beitragen können*. VDI Verlag 2021.

[2] Sauer A, Abele E, Buhl HU. *Energieflexibilität in der deutschen Industrie: Ergebnisse aus dem Kopernikus-Projekt-Synchronisierte und energieadaptive Produktionstechnik zur flexiblen Ausrichtung von Industrieprozessen auf eine fluktuierende Energieversorgung (SynErgie)*. Fraunhofer Verlag 2019.

[3] Reinhart G, Reinhardt S, Graßl M. *Energieflexible Produktionssysteme. Einführungen zur Bewertung der Energieeffizienz von Produktionssystemen*. wt Werkstattstechnik Online 102 2012, 622–628.

[4] Schulz J, Lütkes F, Szabo A, Zaeh MF. *Energy-orientated material flow simulation with stochastic optimisation for peak load management*. Procedia CIRP 107, 2022, 399–404.

[5] Uhlig B, Kloock M, Mennenga M, Herrmann C: *Simulation-based energy flexibility analysis of manufacturing process chains: heat treatment in a foundry*. Procedia CIRP 107, 2022, 1379–1384.

[6] Fecarotta O, Carravetta A, Morani MC, Padulano, R. *Optimal Pump Scheduling for Urban Drainage under Variable Flow Conditions*. Resources, 2018, 7(4), 73.

[7] Luna T, Ribau J, Figueiredo D, Alves R. *Improving energy efficiency in water supply systems with pump scheduling optimization*. Journal of cleaner production. 2019, 342–356.

[8] Fichter C, Müller M. *Intelligente Steuerung von Schöpfwerken zur Reduktion des Energieeinsatzes und der CO2-Emissionen im Projekt Demand Side Management Sielentwässerung–DSMS*. Zeitschrift für Energiewirtschaft 2021, 45(4), 287–294.

[9] Heger J, Voß T. *Simulationsbasierte Optimierung zur Energieersparnis und Verbrauchsflexibilisierung in der Hinterlandentwässerung*. 18. ASIM-Fachtagung Simulation in Produktion und Logistik, Chemnitz, 2019.

[10] Eckhardt K. *Hydrologische Modellierung - Ein Einstieg mithilfe von Excel*. Heidelberg: Springer Spektrum 2014.

[11] Bundesnetzagentur, 2017: *SMARD: Strommarktdaten*. Retrieved January 6, 2023, from https://www.smard.de/home/downloadcenter/download-marktdaten/.

LogFarm: An Open Source Graph-based Simulator for Logistics Networks

Alexander Wuttke[1*], Joachim Hunker[2], Markus Rabe[1]

[1]Department of IT in Production and Logistics, TU Dortmund, Leonhard-Euler-Str. 5, 44227 Dortmund, Germany; *alexander2.wuttke@tu-dortmund.de

[2]Fraunhofer Institute for Software and Systems Engineering, Speicherstr. 6, 44147 Dortmund, Germany;

SNE 34(1), 2024, 43-50, DOI: 10.11128/sne.34.sw.10676
Based on ASIM SPL Simulation Price 2023
Received: 2024-02-29; Accepted: 2024-03-03
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna
ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Logistics networks are complex systems that are often subject to simulation to gain insights. For the computer-aided simulation, simulation tools are used that are also referred to as simulators. In this article, a novel open-source simulator for logistics networks is introduced. The key feature of the tool is its graph-based mode of operation and its usage of simulation models that are stored in graph databases. The simulator is designed to be used in a data farming framework, which means it can be used to generate data along the simulation. The article discusses the general idea and concept behind the authors' graph-based simulator as well as the implementation of an open-source prototype. The prototype is evaluated through the examination of two use cases and a test on the ARGESIM C14 benchmark.

Introduction

Logistics plays a vital role in modern society. The execution of logistics processes, such as transporting, handling, and storing, is performed in logistics systems that consist of multiple interrelated actors. Thus, logistics systems have a network-like structure and are often referred to as logistics networks (LNs). Due to their network-like structure, modeling and simulating LNs as graphs has proven to be a promising approach [1]. It also simplifies the application of graph-based algorithms, e.g., to solve vehicle routing problems as shown in [2].

In order to keep LNs in an efficient and competitive state, simulation is a proven approach at the planning and operational stage. An increasingly popular trend in simulation is to use a simulation model to systematically generate data, which is referred to as data farming [3]. With this approach, data with higher quality can be obtained compared to observational data in real-world applications, which are prone to, e.g., noise and measuring errors [4].

A possible application is to use the data with a subsequent knowledge discovery process, which is referred to as farming-for-mining and is detailed in [5]. Simulation is usually carried out using simulation tools. There is a wide range of different simulation tools available that can simulate LNs, such as AnyLogistix [6] and Enterprise Dynamics [7]. However, those simulation tools rely on relational data and modeling or only partially make use of graphs.

The authors propose to use simulation tools that embrace graph thinking and graph-based approaches, which includes the modeling of simulation models as graphs and the usage of graph algorithms for the simulation of LNs. In this article, the authors point out relevant fundamentals on graphs in general and on labeled property graphs (LPGs) in particular, which are a special type of graph that can be enriched with additional information. Then, key concepts of the novel simulation tool are discussed including the usage in a data farming framework. Based on these concepts, the implementation of an open-source prototype is shown and assessed on the basis of two use cases and tested on the ARGESIM C14 benchmark [8].

1 Related Work

In this section, a concise introduction to the domain of LNs is given and discussed, how simulation as well as data farming are used in this context. Then, key features of graphs and LPGs are highlighted. Also, the mindset and approaches of graph thinking are briefly explained.

1.1 Logistics Networks, Simulation, and Data Farming

LNs are complex systems, consisting of a mesh of different actors such as suppliers, sites, customers, and a multitude of different logistics processes [9]. To keep an LN in a good and competitive state, often the complex interrelationships in an LN are of interest [10]. In this context, simulation has established itself as a proven method for the modeling and analysis of such a complex system [11]. Simulation is defined as the "representation of a system with its dynamic processes in an experimentable model to reach findings, which are transferable to reality; in particular, the processes are developed over time" [12, 3]. The use of a simulation model and efficient experiment design to generate vast amounts of simulation result data is called data farming. According to [3, 1], data farming "is a descriptive metaphor that captures the notion of generating data purposefully to maximize the information yield from simulation models". A procedure model for data farming in a farming-for-mining-framework can be found in [13].

To support a data farming study, simulation tools are used. Simulation tools are a piece of software, consisting of five main components: kernel, data management, user interface, interfaces to external programs, and a predefined modeling world [12, 14]. The modeling world is particularly relevant for the process of building a simulation model. According to [15], the process of building a simulation model is characterized by three consecutive steps: system analysis (resulting in a conceptual model), formalization (resulting in a formal model), and implementation (resulting in an executable model). To guide these steps, a modeling concept is necessary. Established concepts are building blocks (libraries), object-oriented concepts (following the object-oriented paradigm from software development), languages (higher programming languages), and theoretical concepts such as graphs [16].

1.2 Graphs and Graph Thinking

Graph-theoretical modeling concepts are relevant for the representation of complex and interrelated LNs. Although graph theory has been around since Leonhard Euler presented a solution to the problem of the seven bridges of Königsberg, graphs have been gaining interest in research and practice in the last decade [17, 18]. Reasons for this development are advancements in database technology, in particular graph databases, which are a type of NoSQL database system [19]. These developments are driven by the challenges that arise from dependency-oriented and heavily interconnected data, in which explicit interrelations among data are of interest [20]. Graphs are mathematical objects, which naturally describe networks. Typical examples include social networks, public transportation systems, and LNs. Numerous papers report from the successful use of graph-based technologies such as graph databases in logistics, for example, in [21].

Mathematically, a graph is defined as a pair $G = V, E$, with V being a finite set of vertices and E being a finite set of edges. In a logistics context, the term node is used synonymously with vertices. Since the basic structure of nodes and edges cannot represent complex LNs accurately, both can be detailed using properties, in the form of key-value pairs, and labels to identify nodes and edges. This specific type of graph is called an LPG [22], which is a widely used graph model for graph databases, for example, Neo4j [23]. In [24], it is shown how the concept of an LPG can be used to model LNs. An example of an LPG with two nodes, one edge, three labels, and a total of four properties in a logistics application is shown in Figure 1.

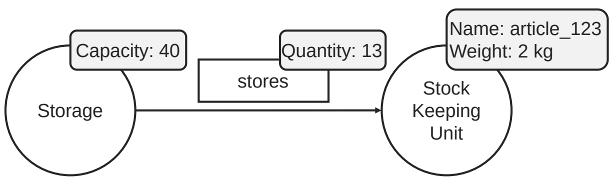


Figure 1: Exemplary labeled property graph.

2 The Simulation Tool

In this section, the novel simulation tool LogFarm developed by the authors of this article is introduced. The simulator is designed graph-based and used in a data farming framework.

The simulation subject is targeted to LNs. The introduction of LogFarm consists of three parts. First, the conceptual model of LNs in LogFarm is presented. This defines what part of the real world can be modeled and simulated with the simulation tool. Second, the general software design are described as well as the tool's functionalities. In the last part, a brief presentation of the implemented prototype is given.

2.1 Conceptual Model of Logistics Networks

The simulation models that can be interpreted and used by the simulator must follow a predefined form. To define the scope of simulation models on a conceptual level, a conceptual model of generic LNs was derived. It serves as a meta model to describe which aspects of LNs can be simulated and at which level of detail. The conceptual model results from a system analysis (see section 2.1).

The level of detail used for the simulation is set at the level of interaction between facilities of actors in the LNs. The considered types of facilities are suppliers, manufacturers, warehouses, and customers. The physical locations of customers, regardless of being an intermediary or end customer, are also referred to as facilities. The simulation progresses over discrete time intervals, each representing a full day. As a consequence, internal processes of facilities and processes that take less than a full day are abstracted and aggregated. Working hours of facilities can be specified for each day of the week, also allowing for disabling a facility for certain days when the working hours are set to zero.

The simulator is capable of simulating transports between facilities of actors and storage processes that take place in LNs. For the transports, vehicle fleets can be assigned consisting of different types of vehicles that can be specified by the user. If a facility distributes goods to multiple other facilities, a dynamic vehicle routing can be used. The geospatial position of locations is specified through the longitude and latitude. The storage keeping is based on policies that can be chosen from a predefined set. Further simulated processes are production and transformation as well as demand and consumption processes that are necessary to establish a flow of goods. The production processes are the sources and the consumption processes are the sinks of goods. Production and transformation processes can optionally be subject to disruptions, also called malfunctions, that are stochastically triggered. The explicitly simulated entities are facilities, goods and vehicles.

2.2 Software Design

The program uses a building-block-oriented modeling concept to create simulation models with regards to the conceptual model, which means that predefined logical entities exist. Each building block resembles a facility of an actor in an LN that acts according to its policy and interacts with other facilities. Therefore, the model can be seen as agent-based.

The building blocks are provided through a building block library that, up to this point, encompasses four different building blocks to model suppliers, manufacturers, warehouses, and customers. Additionally, there is a building block called customer group, which is used to generate large numbers of customers procedurally.

The modeling of a simulation model can take place in the simulation tool itself. LogFarm provides a graphical user interface (GUI) for an editor of simulation models on which, per drag-and-drop, the user can select building blocks from the building block library and place them on a modeling canvas.

Building blocks that should have interactions with other building blocks can also be connected by drag-and-drop of relations. Figure 2 shows the building block library on the left side and on the right side the modeling canvas, where a simple model of two suppliers, one manufacturer, one customer, and three relations can be seen.

Through the GUI of the editor it is also possible to parameterize the buildings blocks after placing them on the modeling canvas. This includes, e.g., setting process rates, defining transformation rules for goods, and enabling malfunctions with certain stochastic traits.

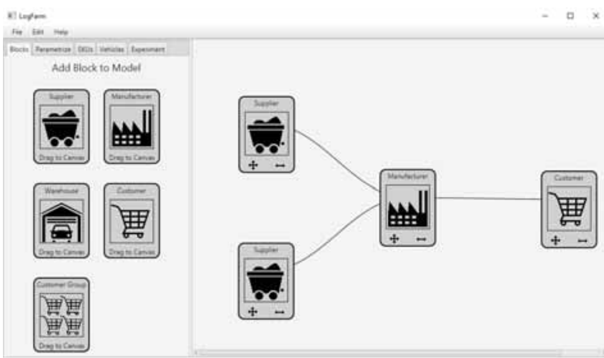


Figure 2: GUI of the editor to build simulation models in the simulation tool.

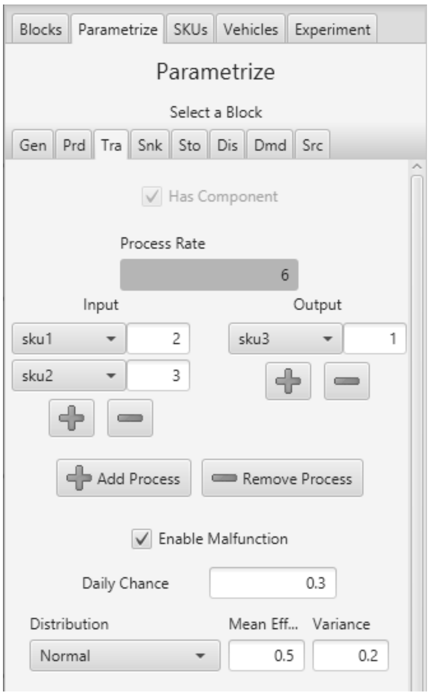


Figure 3: Parametrization window of a block in the simulation tool.

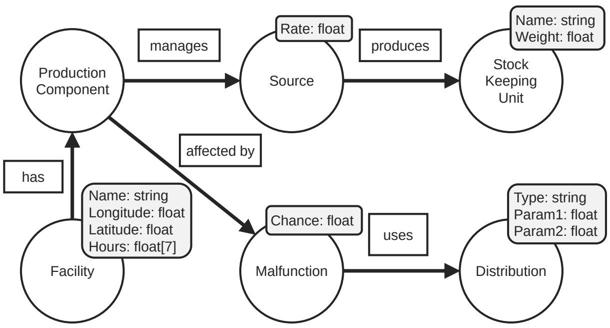


Figure 4: Example of a facility with a production component modeled in an LPG.

Besides giving constant values for parameters, it is also possible to let certain parameters be controlled according to an experiment plan, which stems from a design of experiments and is used when multiple simulation runs are conducted. In this case, the value of the parameter is dynamically changed before a simulation run, according to the value of the corresponding factor in the experiment plan for a given simulation run. To link a parameter to a factor, the identification number of the factor must be entered in the text field and it must be left clicked while holding the CTRL-key.

This will also highlight the text field with a yellow background color to make it distinguishable from constant values. An example of the parametrization window is shown in Figure 3.

After finishing modeling a simulation model in the editor it can be saved to a graph database in form of an LPG. It is also possible to load a simulation model in the form of an LPG and simulate it.

The modeling of a simulation model as an LPG is guided by a modeling framework that is only briefly described in this article. A more in-depth explanation is provided in prior work of the authors [24]. The modeling framework consists of a set of labels, a set of possible properties, rules for edges, and general modeling rules. In Figure 4, a supplier with its production component affected by malfunctions is modeled, which will serve as an explanatory example. In the case of LogFarm, properties are only attached to nodes. For example, facilities can be detailed by a name, geo-coordinates, and working hours. Those properties correspond to the parameters of the building blocks. Because the facility node is connected to a production component node, the simulator recognizes that the facility is able to execute production processes. What is to be produced is specified by the connected source node and stock keeping unit node. The simulator can also recognize that the production component is affected by malfunctions that occur by a given probability and have a failure duration determined stochastically by a given distribution. The shown example is only a small part of a valid simulation model that can be used for LogFarm. Valid simulation models can feature a much more complex and rich structure of nodes, edges, and properties.

Another functionality of LogFarm is to read externally generated experiment plans for simulation. The experiment plan must be in the CSV file format and can be generated by tools such as MATLAB [25].

Regarding the parameters of the simulation itself, LogFarm supports the specification of a simulation start and end date as well as the definition of a warm-up phase, in which no data are logged or generated. It is also possible to specify a seed to fix the sequence of random numbers and, therefore, enable reproducibility of simulation runs.

Given a simulation model, an experiment plan, and simulation parameters, the actual simulation can take place. The simulation is controlled by an experiment manager, whose task is to execute all simulation runs of the experiment plan.

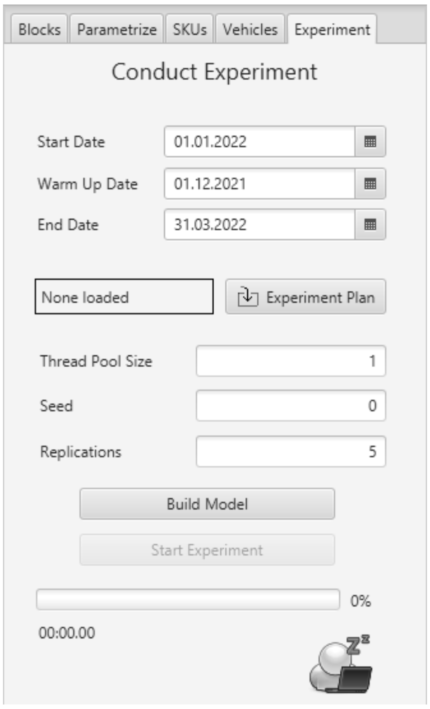


Figure 5: Experiment window of a block in the simulation tool.

The experiment manager of LogFarm is capable of parallel execution of simulation runs. For now, this is limited to the kernels and threads of one local machine, but can be adapted to work in distributed systems. The simulation is carried out with the simulation kernel and yields simulation result data. The GUI to enter simulation parameters and to control the experiment manager is shown in Figure 5.

Usually, simulation is used to gain insights. As described in in Section 2.1, simulation can also be used to generate data along the simulation in data farming frameworks. LogFarm is intended to be used in a data farming framework and is, therefore, capable of generating data.

Currently, data on stocks and orders are generated and perpetually written to the database. This enables the detailed analysis of transactional data in the LNs after simulation with, e.g., a farming-for-mining-framework. In the future, more data of interest can be considered and generated. An overview of the software architecture of LogFarm is shown in Figure 6.

2.3 Implementation

LogFarm is implemented as an open-source simulator programmed in Java. The source code is hosted on github in the repository found at https://github.com/Alexw1011/LogFarm.

The GUI of LogFarm is implemented in JavaFX. For the graph database, Neo4j is used (see Section 2.2). Neo4j is a well-known graph database system that uses an LPG as data model and supports object graph mapping for efficient read and write operations. In Figure 7, a small simulation model for LogFarm in the form of an LPG is visualized by Neo4j.

3 Evaluation

To validate the introduced concepts and test the developed prototype, results of two recent case studies are presented. Furthermore, the authors have implemented the ARGESIM C14 benchmarks for comparability.

3.1 Case Studies

The prototype was tested on two case studies. One is a typical LN consisting of three suppliers, one manufacturer, one warehouse, and three customers. The other case study is based on a real use case of city logistics on the metropolitan area of Athens, which was examined in a prior study using a different simulation tool [26]. In this case study, a two-echelon LN was examined that featured five suppliers, one consolidation center, and 8,537 customers.

For more information on the case studies, the reader is kindly referred to [27]. The main takeaways from these case studies are that the bottleneck is often not the time for the simulation itself, but the write operations to the graph database, which can be higher than the simulation time. This is especially problematic when using parallel simulation but only one graph database to write to, as a considerable amount of time cannot be used for simulation but is spent to wait for the graph database write operations.

3.2 Benchmarks

The following benchmark was conducted accordingly to the ARGESIM C14 benchmark. It is a simplified two-echelon LN with four factories, four distributors, a group of wholesalers, and twelve products.

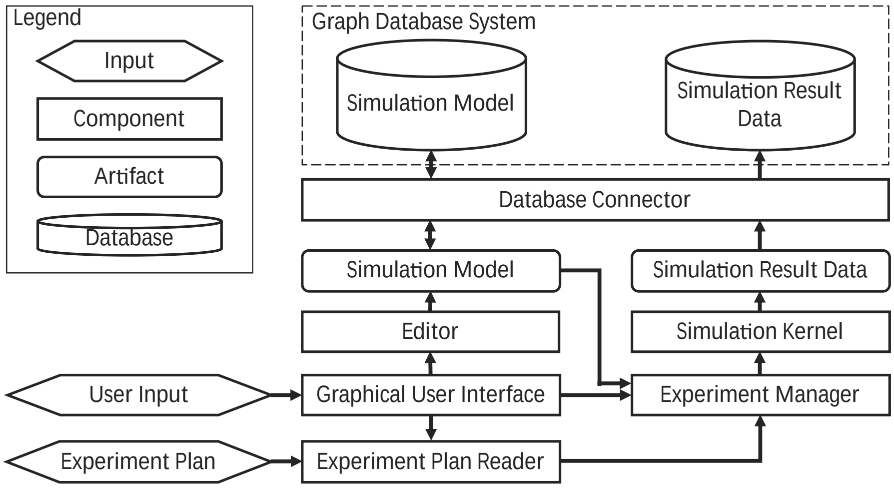


Figure 6: Software architecture of the simulation tool.

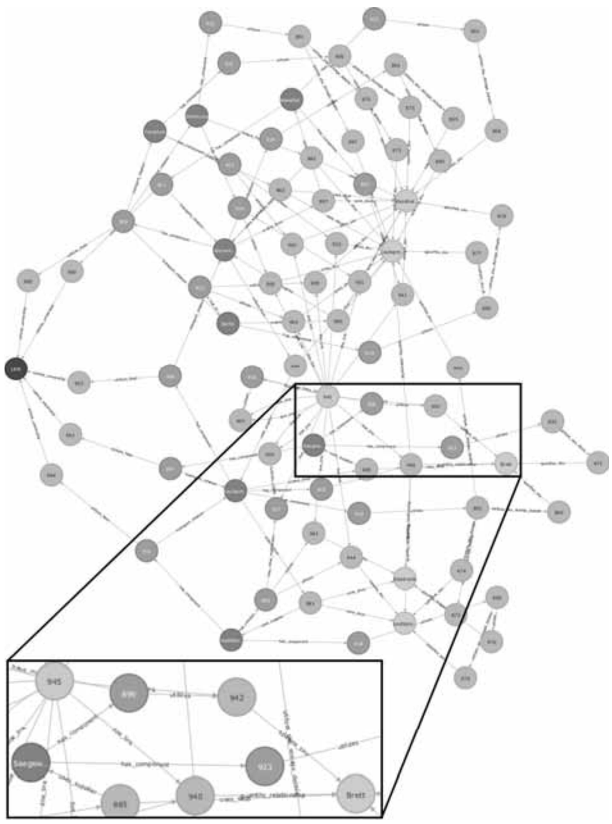


Figure 7: Exemplary simulation model in the Neo4j GUI.

A total of six tasks is considered with different order strategies of the distributors and wholesalers. Following these, the details of the benchmark can be found in [8]. A model has been created and its parameters varied for the different tasks. To model the factories, a total of four supplier building blocks are used. The four distribution centers are modeled with one warehouse building block each. The group of wholesalers is represented by one customer building block. A screenshot of the model in LogFarm can be seen in Figure 8.

The results are presented in the Tables 1, 2, and 3, as well as Figures 9 and 10. Compared to benchmark results obtained by MATLAB [28] and Enterprise Dynamics [29], the results are similar with only minor deviations, which are likely due to stochastic influences. Based on this comparison, it can be concluded that LogFarm can achieve equivalent results to established simulation tools in the field of LNs.

	C	N	R
min	28,441	190	139.2
max	35,012	253	162.8
mean	32,866	223	148.1
deviation	1,320	12.2	6.1

Table 1: Benchmark results of ARGESIM C14 task a2.

4 Conclusion and Outlook

In this paper, an open source graph-based simulator for LNs called LogFarm is presented. In particular, the concept of LPGs is used, which has proven to be an intuitive and appropriate way to model LNs for simulation.

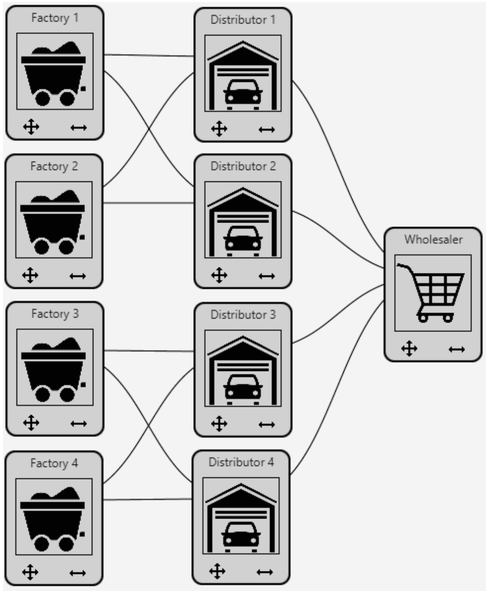


Figure 8: Screenshot of the model used for the ARGESIM C14 benchmark.

	C	N	R
min	27,532	193	129.3
max	34,090	253	157.2
mean	32,021	224	143.0
deviation	1,217	12.0	5.8

Table 2: Benchmark results of ARGESIM C14 task b2.

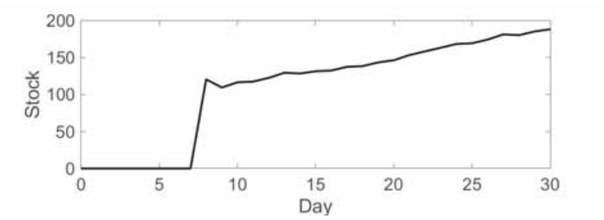


Figure 9: Results for the stock in ARGESIM C14 task a1.

Its applicability is shown in two case studies and in the ARGESIM C14 benchmark.

The authors' goal is to stimulate a more extensive use of graph thinking. The introduced simulation tool is a proof of concept, which has shown very promising results and already sparked interest in various new approaches. Although the shown prototype of LogFarm is capable of simulating many typical scenarios of LNs, the range of functions offered is still limited compared

	C	N	R
min	22,882	192	106.9
max	29,980	251	141.5
mean	26,798	224	119.6
deviation	1,122	12.3	5.1

Table 3: Benchmark results of ARGESIM C14 task c1.

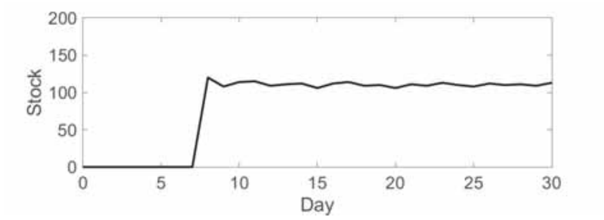


Figure 10: Results for the stock in ARGESIM C14 task b1.

to established simulation tools. Therefore, the next step to increase the maturity level of the software is to add more functionalities like, e.g., distance calculations not only based on geo-coordinates, more options to model stochastic influences, and more realistic ways to generate demand of customers as suggested in [30].

References

[1] Domschke W, Drexl A, Klein R, Scholl A. *Einführung in Operations Research*. Wiesbaden, Germany: Springer Gabler. 2015.

[2] Barceló J, Grzybowska H, Pardo S. Vehicle Routing and Scheduling Models, Simulation and City Logistics: 2: 3. In: *Dynamic Fleet Management: 10*, edited by Zeimpekis V, Tarantilis CD, Giaglis GM, Minis I, pp. 163–195. Boston, MA, USA: Springer US. 2007.

[3] Sanchez S. Data Farming: Methodes for the Present, Opportunities for the Future. *ACM Transactions on Modeling and Computer Simulation*. 2020;30(4):1–30.

[4] García S, Luengo J, Herrera F. *Data Preprocessing in Data Mining*. Cham: Springer International Publishing. 2015.

[5] Hunker J, Wuttke A, Scheidler AA, Rabe M. A Farming-for-Mining-Framework to Gain Knowledge in Supply Chains. In: *Proceedings of the 2021 Winter Simulation Conference*, edited by Kim S, Feng B, Smith K, Masoud S, Zheng Z, Szabo C, Loper M. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Inc. 2021; pp. 1–12.

[6] The Anylogic Company. AnyLogistix. 2024. URL https://www.anylogistix.com/

[7] InControl. Enterprise Dynamics. 2024. URL https://www.incontrolsim.com/software-platform/

[8] Tauböck S. C14 Supply Chain Management - Definition. *SNE Simulation Notes Europe*. 2001; 11(32/33):42–43.

[9] Christopher M. *Logistics & Supply Chain Management*. Harlow, England: Pearson Education, 5th ed. 2016.

[10] Pfohl HC. *Logistics Systems: Business Fundamentals*. Wiesbaden, Germany: Springer Gabler. 2022.

[11] Law A. *Simulation Modeling and Analysis*. NY, USA: McGraw-Hill Education, 5th ed. 2015.

[12] VDI-Guideline 3633 Part 1. Simulation of Systems in Materials Handling, Logistics and Production: Fundamentals. 2014.

[13] Hunker J, Scheidler AA, Rabe M, van der Valk H. A New Data Farming Procedure Model for a Farming for Mining Method in Logistics Networks. In: *Procedings of the 2022 Winter Simulation Conference*, edited by Feng B, Pedrielli G, Peng Y, Shashaani S, Song E, Corlu CG, Lee LH, Chew EP, Roeder T, Lendermann P. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc. 2022; pp. 1461–1472.

[14] Wenzel S. Simulation logistischer Systeme. In: *Modellierung logistischer Systeme*, edited by Tempelmeier H, pp. 1–34. Wiesbaden, Germany: Springer Gabler. 2018.

[15] Rabe M, Spieckermann S, Wenzel S. *Verifikation und Validierung für die Simulation in Produktion und Logistik: Vorgehensmodelle und Techniken*. Berlin, Germany: Springer. 2008.

[16] Gutenschwager K, Rabe M, Spieckermann S, Wenzel S. *Simulation in Produktion und Logistik*. Berlin, Heidelberg, Germany: Springer. 2017.

[17] Jungnickel D. *Graphs, Networks and Algorithms*. Berlin, Germany: Springer, 4th ed. 2013.

[18] Robinson I, Webber J, Eifrem E. *Graph Databases: New Opportunities for Connected Data*. Sebastopol, CA: O'Reilly, 2nd ed. 2015.

[19] Meier A, Kaufmann M. *SQL & NoSQL Databases: Models, Languages, Consistency Options and Architectures for Big Data Management*. Wiesbaden, Germany: Springer Fachmedien. 2019.

[20] Aggarwal CC. *Data Mining*. Cham: Springer International Publishing. 2015.

[21] Xiaoliang Z, Qingtao Z, Mingjie T, Hui H. Design and Implementation of Express Logistics System Based on Graph Database and Baidu Map. In: *Proceedings of the 2nd International Conference on Information Technology and Computer Application*, edited by Freris N, Bin Zakaria Z. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Inc. 2020; pp. 396–400.

[22] Angles R. The Property Graph Database Model. In: *Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management*, edited by Olteanu D, Poblete B. Aachen, Germany: CEUR-WS. 2018; pp. 16–26.

[23] Neo4j. Neo4j. 2024. URL https://neo4j.com/

[24] Wuttke A, Hunker J, Rabe M. Modeling of Logistics Networks with Labeled Property Graphs for Simulation in Digital Twins. In: *Proceedings of the EUROSIM Congress 2023*, edited by Mota M, Mendoza A, Scala P. Cham: Springer. 2023; Accepted, to be published.

[25] MathWorks. MATLAB. 2024. URL https://de.mathworks.com/products/matlab.html

[26] Rabe M, Klueter A, Wuttke A. Evaluating the Consolidation of Distribution Flows Using a Discrete Event Supply Chain Simulation Tool: Application to a Case Study in Greece. In: *Proceedings of the 2018 Winter Simulation Conference*, edited by Rabe M, Juan AA, Mustafee A, Skoogh SJ, Johansson B. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc. 2018; pp. 2815–2826.

[27] Wuttke A. Entwicklung eines graphbasierten Simulators zur Simulation von Logistiknetzwerken in einem Data-Farming-Framework. Master Thesis TU Dortmund, ITPL. 2022. URL https://www.itpl-vw.mb.tu-dortmund.de/publikationen/MA_2022_Wuttke.pdf

[28] Wastian M, Reichl S. An Object-oriented Approach to ARGESIM Benchmark C14 'Supply Chain' using MATLAB. *SNE Simulation Notes Europe*. 2018; 28(1):35–38.

[29] Gutwenger P, Hoa DD, Puffitsch W, Tauböck S. A Scipt Language-supported Approach to ARGESIM Benchmark C14 'Supply Chain' in Enterprise Dynamics. *SNE Simulation Notes Europe*. 2011; 21(1):55–56.

[30] Wuttke A, Hunker J, Scheidler AA, Rabe M. Synthetic Demand Generation with Seasonality for Data Mining on a Data-Farmed Data Basis of a Two-Echelon Supply Chain. *Procedia Computer Science*. 2022;204:226–234.

EUROSIM Data and Quick Info

ASIM SST 2024 — **27. Symposium Simulationstechnik**
27th Symposium Simulation Technique
4. - 6. September 2024, München, Univ. BW, Deutschland www.asim-gi.org

SIMS EUROSIM 2024 – 2nd SIMS EUROSIM Conference on Modelling and Simulation, with SIMS 2024 – 65th International Conference of Scandinavian Simulation Society
Oulu, Finland, September 10-12, 2024 www.scansims.org

SIMS

I3M 2024 International Multidisciplinary Modeling & Simulation Multiconference
Tenerife, Spain, September 18-20, 2024
www.msc-les.org/i3m2024

Winter Simulation Conference 2024, December 15-18, 2024
Orlando, FL, USA www.wintersim.org

MATHMOD 2025 11th Vienna Int. Conference on Mathematical Modelling
February 19 – 21 , 2025, Vienna, Austria
www.mathmod.at

EUROSIM CONGRESS 2026
July 2026, Italy
www.eurosim.info

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 modelling and simulation in industry, research, and development – by publication and conferences.

www.eurosim.info

EUROSIM members may be national simulation societies and regional or international societies and groups dealing with modelling and simulation.

Full Members are ASIM, CEA-SMSG, CSSS, DBSS, KA-SIM, LIOPHANT, LSS, PTSK, NSSM, SIMS, SLOSIM, UKSIM. Observer Members are ALBSIM and ROMSIM. Former Members (societies in re-organisation) are: CROS-SIM, FRANCOSIM, HSS, ISCS.

EUROSIM is governed by a B**oard** consisting of one representative of each member society, president, past president, and **SNE** representative.

President	Agostino Bruzzone (LIOPHANT) *agostino@itim.unige.it*
Past President	M. Mujica Mota (DBSS), *m.mujica.mota@hva.nl*
Secretary	Marina Massei (LIOPHANT), *massei@itim.unige.it*
Treasurer	Felix Breitenecker (ASIM) *felix.breitenecker@tuwien.ac.at*
Webmaster	Irmgard Husinsky (ASIM), *irmgard.husinsky@tuwien.ac.at*

SNE – Simulation Notes Europe is EUROSIM's membership journal with peer reviewed scientific contributions about all areas of modelling and simulation, including new trends as big data, cyber-physical systems, etc.

The EUROSIM societies distribute **e-SNE** in full version to their members as official membership journal. The basic version of **e-SNE** is available with open access. Publishers are **EUROSIM**, **ARGESIM** and **ASIM**,

www.sne-journal.org

office@sne-journal.org

SNE-Editor: Felix Breitenecker (ASIM)
felix.breitenecker@eic@sne-journal.org

EUROSIM Congress and Conferences

Each year a major **EUROSIM** event takes place, as the **EUROSIM CONGRESS** organised by a member society, **SIMS EUROSIM Conference**, and **MATHMOD Vienna Conference (ASIM)**.

On occasion of the **EUROSIM Congress 2023**, the *11th EUROSIM Congress* in Amsterdam, July, 2023, a new **EUROSIM** president has been elected: we welcome Agostino Bruzzone, well known simulationist, as new president. His society **LIOPHANT** will organize the next EUROSIM Congress in 2026 in Italy.

Furthermore, **EUROSIM Societies** organize local conferences, and **EUROSIM** co-operates with the organizers of **I3M Conference** and **WinterSim Conference Series.**

EUROSIM Member Societies

ASIM German Simulation Society Arbeitsgemeinschaft Simulation

ASIM is the association for simulation in the German speaking area, servicing mainly Germany, Switzerland and Austria.

President	Felix Breitenecker, *felix.breitenecker@tuwien.ac.at*
Vice President	Sigrid Wenzel, *s.wenzel@uni-kassel.de* Thorsten Pawletta, *thorsten.pawletta@hs-wismar.de* Andreas Körner, *andreas.koerner@tuwien.ac.at*

ASIM is organising / co-organising the following international conferences: **ASIM SPL** Int. Conference 'Simulation in Production and Logistics' (biannual), **ASIM SST** 'Symposium Simulation Technique' (biannual), **MATHMOD** Int. Vienna Conference on Mathematical Modelling (triennial). Furthermore, **ASIM** is co-sponsor of *WSC* - Winter Simulation Conference and of the *I3M* and conference series.

ASIM Working Committees

GMMS: Methods in Modelling and Simulation
U. Durak, umut.durak@dlr.de
SUG: Simulation in Environmental Systems
J. Wittmann, wittmann@informatik.uni-hamburg.de
STS: Simulation of Technical Systems
W. Commerell, commerell@hs-ulm.de
SPL: Simulation in Production and Logistics
S. Wenzel, s.wenzel@uni-kassel.de
EDU: Simulation and Education
A. Körner, andreas.koerner@tuwien.ac.at
Working Group Big Data: Data-driven Simulation in Life Sciences, N. Popper, niki.popper@dwh.at
Other Working Groups: Simulation in Business Administration, in Traffic Systems, for Standardisation, etc.

Contact Information

www.asim-gi.org

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

ASIM – Inst. of Analysis and Scientific Computing, TU Wien, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria

CEA-SMSG – Spanish Modelling and Simulation Group

CEA is the Spanish Society on Automation and Control. The association is divided into national thematic groups, one of which is centered on Modeling, Simulation and Optimization (**CEA-SMSG**).

President	José L. Pitarch, *jlpitarch@isa.upv.es*
Vice President	Juan Ignacio Latorre, *juanignacio.latorre@unavarra.es*

Contact Information

www.ceautomatica.es/grupos/

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

CSSS – Czech and Slovak Simulation Society

CSSS is the Simulation Society with members from the two countries: Czech Republic and Slovakia. The CSSS history goes back to 1964.

President	Michal Štepanovský *michal.stepanovsky@fit.cvut.cz*
Vice President	Mikuláš Alexík, *alexik@frtk.fri.utc.sk*

Contact Information

cssim.cz

michal.stepanovsky@fit.cvut.cz

CSSS – Český a Slovenský spolek pro simulaci systémů, Novotného lávka 200/5,
11000 Praha 1, Česká republika

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.

President	M. Mujica Mota, *m.mujica.mota@hva.nl*
Vice President	A. Heemink, *a.w.heemink@its.tudelft.nl*
Secretary	P. M. Scala, *paolo.scala@fedex.com*

Contact Information

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

KA-SIM Kosovo Simulation Society

The Kosova Association for Modeling and Simulation (**KA-SIM**) is closely connected to the University for Business and Technology (UBT) in Kosovo.

President	Edmond Hajrizi, *ehajrizi@ubt-uni.net*
Vice President	Muzafer Shala, *info@ka-sim.com*

Contact Information

www.ubt-uni.net

ehajrizi@ubt-uni.net

Dr. Edmond Hajrizi
Univ. for Business and Technology (UBT)
Lagjja Kalabria p.n., 10000 Prishtina, Kosovo

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.

President	A.G. Bruzzone, *agostino@itim.unige.it*
Director	E. Bocca, *enrico.bocca@liophant.org*

Contact Information

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

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.

President	Artis Teilans, *Artis.Teilans@rta.lv*
Vice President	Oksana Kuznecova, *Oksana.Kuznecova@rtu.lv*

Contact Information

www.itl.rtu.lv/imb/

Artis.Teilans@rta.lv, Egils.Ginters@rtu.lv

LSS, Dept. of Modelling and Simulation, Riga Technical University, Kalku street 1, Riga, LV-1658, Latvia

NSSM – National Society for Simulation Modelling (Russia)

NSSM – The National Society for Simulation Modelling (Национальное Общество Имитационного Моделирования – НОИМ) was officially registered in Russia in 2011.

President	R. M. Yusupov, *yusupov@iias.spb.su*
Chairman	A. Plotnikov, *plotnikov@sstc.spb.ru*

Contact Information

www.simulation.su

yusupov@iias.spb.su

NSSM / R. M. Yusupov, St. Petersburg Institute of Informatics and Automation RAS, 199178, St. Petersburg, 14th line, h. 39

PTSK – Polish Society for Computer Simulation

PTSK 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.

President	Tadeusz Nowicki, *Tadeusz.Nowicki@wat.edu.pl*
Vice President	Leon Bobrowski, *leon@ibib.waw.pl*

Contact Information

www.ptsk.pl

leon@ibib.waw.pl

PSCS, 00-908 Warszawa 49, ul. Gen. Witolda Urbanowicza 2, pok. 222

SIMS – Scandinavian Simulation Society

SIMS is the Scandinavian Simulation Society with members from the five Nordic countries Denmark, Finland, Norway, Sweden and Iceland. The **SIMS** history goes back to 1959.

President	Tiina Komulainen, *tiina.komulainen@oslomet.no*
Vice President	Erik Dahlquist, *erik.dahlquist@mdh.se*

Contact Information

www.scansims.org

vadime@wolfram.com

Vadim Engelson, Wolfram MathCore AB,
Teknikringen 1E, 58330, Linköping, Sweden

SLOSIM – Slovenian Society for Simulation and Modelling

The Slovenian Society for Simulation and Modelling was established in 1994. It promotes modelling and simulation approaches to problem solving in industrial and in academic environments by establishing communication and cooperation among corresponding teams.

President	Goran Andonovski, *goran.andonovski@fe.uni-lj.si*
Vice President	Božidar Šarler, *bozidar.sarler@fs.uni-lj.si*

Contact Information

www.slosim.si

slosim@fe.uni-lj.si, vito.logar@fe.uni-lj.si

SLOSIM, Fakulteta za elektrotehniko, Tržaška 25,
SI-1000, Ljubljana, Slovenija

UKSIM - United Kingdom Simulation Society

The UK Modelling & Simulation Society (**UKSim**) is the national UK society for all aspects of modelling and simulation, including continuous, discrete event, software and hardware.

President	David Al-Dabass, *david.al-dabass@ntu.ac.uk*
Secretary	T. Bashford, *tim.bashford@uwtsd.ac.uk*

Contact Information

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

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.

Contact Information

florin_h2004@yahoo.com
ROMSIM / Florin Hartescu, National Institute for Research in Informatics, Averescu Av. 8 – 10, 011455 Bucharest, Romania

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.

Contact Information

kozeta.sevrani@unitir.edu.al

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

Former Societies / Societies in Re-organisation

- **CROSSIM** – *Croatian Society for Simulation Modelling*
  Contact: Tarzan Legović, *Tarzan.Legovic@irb.hr*
- **FrancoSim** – Société Francophone de Simulation
- **HSS** – Hungarian Simulation Society
  Contact: A. Gábor, *andrasi.gabor@uni-bge.hu*
- **ISCS** – Italian Society for Computer Simulation

The following societies have been formally terminated:

- **MIMOS** –Italian Modeling & Simulation Association; terminated end of 2020.

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 **ARGESIM 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
- Support of the Conference Series **MATHMOD Vienna** (triennial, in co-operation with **EUROSIM**, **ASIM**, and **TU Wien**) – www.mathmod.at
- Administration of **ASIM** (German Simulation Society) and administrative support for **EUROSIM** *www.eurosim.info*
- Simulation activities for TU Wien

ARGESIM is a registered non-profit association and a registered publisher: **ARGESIM Publisher Vienna**, root ISBN 978-3-901608-xx-y and 978-3-903347-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.

SNE Editorial Office /ARGESIM

→**www.sne-journal.org**

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

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

SNE, primarily an electronic journal, follows an open access strategy, with free download in a basic version (B/W, low resolution graphics). **SNE** is the official membership journal of **EUROSIM**, the *Federation of European Simulation Societies*. Members of (most) **EUROSIM Societies** are entitled to download the full version of e-**SNE** (colour, high-resolution graphics), and to access additional sources of benchmark publications, model sources, etc. (group login for the ‘publication-active’ societies; please contact your society). Furthermore, **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:

→ **www.sne-journal.org**,