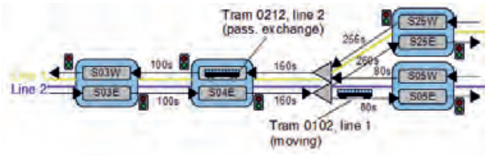
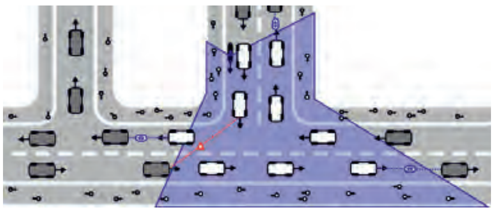
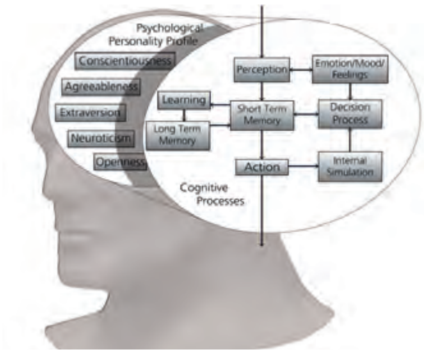
SNE SIMULATION NOTES EUROPE

SNE Special Issue

Simulation of Traffic Systems - Technical Systems







Volume 22 Number 2
August 2012

Print ISSN 2305-9974
Online ISSN 2306-0271

Journal on Developments and
Trends in Modelling and Simulation

Membership Journal for Simulation
Societies and Groups in EUROSIM

ARGESIM


515.000.000 KM, 380.000 SIMULATIONEN
UND KEIN EINZIGER TESTFLUG.
DAS IST MODEL-BASED DESIGN.
Nachdem der Endabstieg der beiden Mars Rover unter Tausenden von atmosphärischen Bedingungen simuliert wurde, entwickelte und testete das Ingenieur-Team ein ausfallsicheres Bremsraketen-System, um eine zuverlässige Landung zu garantieren. Das Resultat – zwei erfolgreiche autonome Landungen, die exakt gemäß der Simulation erfolgten.
Mehr hierzu erfahren Sie unter:
www. mathworks.de/mbd
MATLAB®
&SIMULINK®
The MathWorks
Accelerating the pace of engineering and science

SNE SPECIAL ISSUE Simulation of Traffic Systems - Technical Systems

Modeling and simulation of traffic systems has a long tradition in EUROSIM/ASIM, and researchers are organized in a special ASIM expert group. In recent years there has been a generation change in this field and many of the mostly young researchers are interested in a broader context, particularly the basic principles of modeling and simulation. Therefore, the former autonomous expert group was integrated as a working group in the ASIM group for Fundamentals and Methods of Modeling and Simulation (GMMS) early 2012. This special issue presents primary contributions of young researchers in the new working group.

The SNE special issue on *Simulation of Traffic Systems / Technical Systems* is structured into eight contributions, where some contributions demonstrate specific project results, while others discuss new project ideas.

The first two contributions present parts of a project on simulation and optimization of tram schedules that is processed at the University of Cologne. Lückerath et al. describe the development of a simulation model to study time-table-based tram traffic. Ullrich et al. discuss the problem of tram schedules in a broader context and present a model-based tool chain to generate and evaluate tram schedules. Both contributions apply their research to Cologne's tram network and present some results.

The contribution on a simulation based approach on robust airline job pairing by Kuckertz et al. describes a project that is under development. Its objective is the model-based generation of robust personnel schedules for airline operations.

The fourth paper on cognitive aspects of traffic simulations in virtual environments by Seele et al. presents a concept for traffic simulation with regard to road safety education. It is based on modeling persistent traffic agents whose behaviour is inspired by human cognitive processes.

The contribution on network-wide evaluation of cooperative traffic systems using microscopic traffic flow simulation by Lüßmann and Santa describe the concept of a traffic flow simulation environment and its integration in the European eCoMove project. M. Bruckner's contribution on student pedestrian traffic deals with the relatively new area of pedestrian traffic simulation, using cellular automata for spatial movements in buildings.

The last two contributions come from the area of technical systems. The paper by Braig et al. briefly introduces the Modelica library 'AlternativeVehicles' and demonstrates its use with the example of modeling the fuel consumption of a conventional vehicle and a parallel hybrid vehicle. Schwatinski's contribution on robot control introduces a new concept for flexible task management based on system entity structure.

The editors would like to thank Oliver Ullrich and his colleagues Daniel Lückerath and Patrick Kuckertz from the University of Cologne in Germany for their help in organizing this special issue, and Felix Breitenecker, editor-in-chief of SNE, for the opportunity for this special issue. Furthermore, the editors would like to express their gratitude to all authors for their cooperation and efforts, e.g. for sending revised versions. We hope that the selected papers present a good overview of current work in this research area.

Thorsten Pawletta, Christina Deatcu
Wismar University of Applied Sciences, Germany,
thorsten.pawletta@hs-wismar.de

Contents

Editorial

Dear Readers – *This SNE Special Issue on 'Simulation of Traffic Systems - Technical Systems' proves the benefits of modelling and simulation for solving nowadays and future problems also in the area of traffic systems. While classical areas of modelling and simulation work with genuine approaches, traffic system modelling and simulation requires use of very different approaches, from ODEs to cellular automata, from DEVS to agent-based – and what makes the tasks exciting – unusual combinations of all these approaches. We are glad, that our new SNE can publish a special issue on this fascinating area.*

Our new SNE – Simulation News Europe - comes along with DOI indexing and with open access strategy for Online SNE, a demand and need especially from and for young simulationists, as for authors of this issue.

I would like to thank all authors and all people who helped in managing this SNE Special Issue, especially the Issue Editors Mrs. Christina Deatcu and Mr. Thorsten Pawletta, Wismar University of Applied Science.

Felix Breitenecker, Editor-in-Chief, 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 and handling of *Technical Notes*, *Education Notes*, *Short Notes*, *Software Notes*, *Overview Notes*, and of *Benchmark Notes* (definitions and solutions). At present, the board is increasing:

SNE Contact & Info

→ *www.sne-journal.org*
office@sne-journal.org, eic@sne-journal.og
SNE Editorial Office, c/o ARGESIM / DWH,
Neustiftgasse 57-59, 1070 Vienna, Austria

SNE SIMULATION NOTES EUROPE

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

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

Scope: Technical Notes, Short Notes and Overview Notes on developments and trends in modelling and simulation in various areas and in application and theory; benchmarks and benchmark documentations of ARGESIM Benchmarks on modelling approaches and simulation implementations; modelling and simulation in and for education, simulation-based e-learning; society information and membership information for EUROSIM members (Federation of European Simulation Societies and Groups).

Editor-in-Chief: Felix Breitenecker, Vienna Univ. of Technology, Inst. f. Analysis and Scientific Computing, Div., Math. Modelling and Simulation, Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria; *Felix.Breitenecker@tuwien.ac.at*, *eic@sne-journal.org*

Layout / Administration: Anna Mathe, ARGESIM /ASIM *anna.mathe@tuwien.ac.at*, *office@sne-journal.org*

Print SNE: Grafisches Zentrum, TU Vienna, Wiedner Hauptstrasse 8-10, 1040, Vienna, Austria

Online SNE: DWH Simulation Services, Neustiftgasse 57-59, 1070 Vienna

Publisher: ARGESIM ARBEITSGEMEINSCHAFT SIMULATION NEWS - WORKING COMMITTEE SIMULATION NEWS, Neustiftgasse 57-59, 1070 Vienna, Austria; → *www.argesim.org*, *info@argesim.org* on behalf of ASIM(→ *www.asim-gi.org* and EUROSIM → *www.eurosim.info*

Modeling Time Table based Tram Traffic

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

[1]University of Cologne, Inst. f. Informatics, Pohligstrasse 1, 50969 Cologne, Germany;
[2]Cologne University of Applied Science, Colohne, Germany; **lueckerath@informatik.uni-koeln.de*

Abstract. In mid-sized cities, tram networks are major components of public service infrastructure. In those networks with their typically dense schedules, multiple lines share tracks and stations, resulting in a dynamic system behavior and mounting delays following even small disruptions. Robustness is an important factor to keep delays from spreading through the network and to minimize average delays.

This paper describes part of a project on simulation and optimization of tram schedules, namely the development and application of a simulation model representing a tram network and its assigned time table. We begin by describing the components of a tram network, which consist of physical and logical entities. These concepts are then integrated into a model of time table based tram traffic. We apply the resulting simulation software to our hometown Cologne's tram network and present some experimental results.

Introduction

Tram networks are important parts of public transport infrastructure, which is exemplified by the 745,000 passengers that are transported in Cologne's tram network every day as described in [5]. Especially mid-sized cities often have mixed tram networks, i.e. networks where trams travel on street level (thus being subject to individual traffic and corresponding traffic regulation strategies) and on underground tracks. In such dense networks robustness is an important factor to minimize average delays. Robustness measures the degree on which inevitable small disturbances, e.g. obstructed tracks due to parked cars, have impact on the whole network. With robust time tables delays are kept at a local level, whereas with non-robust time tables they spread through the network and might subsequently cause delays of other vehicles as described in [2, 3].

In this paper we present the simulation module (first described in [4]) which is part of a larger project to generate and evaluate robust time tables in order to minimize the impact of small delays. We develop a model and implement an application to simulate time tables of mixed tram networks in order to evaluate given time tables before their implementation in the field and to compare time tables generated by optimization methods (as in [7, 8]) with respect to their applicability. In addition we want to show that the adopted simulation engine can be applied to real world problems.

A more detailed description of our project and in particular our optimization approach is presented in the accompanying paper ‘Simulation and optimization of Cologne's tram schedule’ (see [7]).

We begin the remainder of this paper by describing the basics of time table based tram traffic (Section 1), followed by a short discussion of our model representing the physical and logical entities of the tram network (Section 2). The resulting software is then applied to Cologne's tram network (Section 3). We close with a short summary of the lessons learned and give some remarks on further research (Section 4).

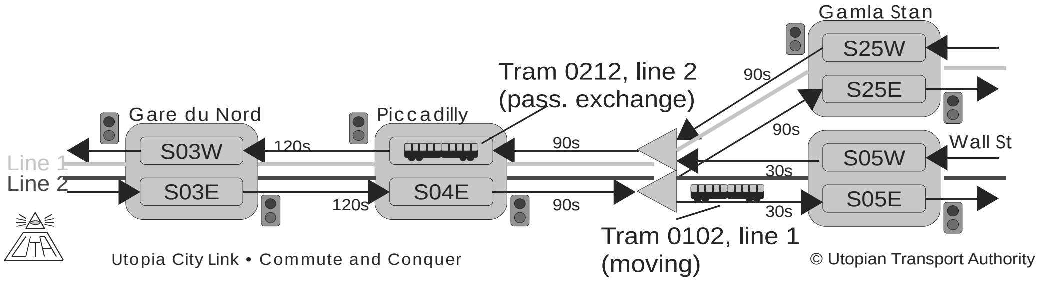


Figure 1. Part of a tram network.

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 22(2), 2012, 61-68 | doi: 10.11128/sne.22.tn.10121

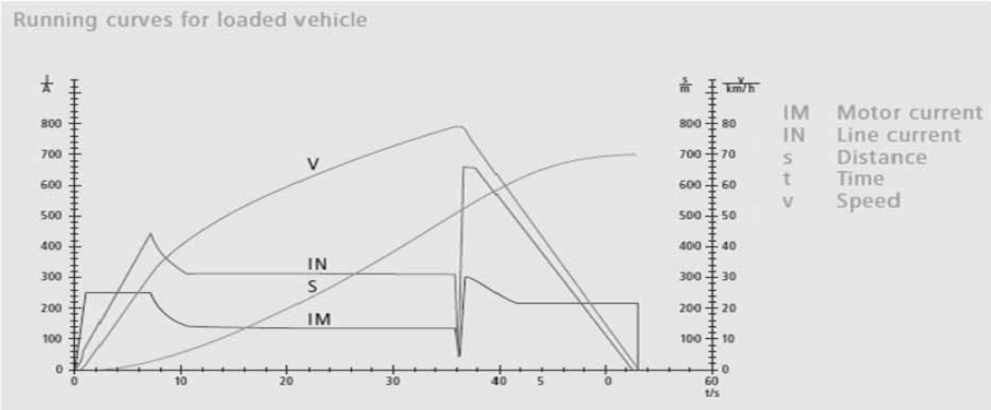


Figure 2. Maneuvering capabilities of wagon type K4000 as found in [10].

1 Time Table - based Tram Traffic

Tram networks can be considered as a combination of physical and logical components. The physical network consists of tangible entities, e.g. stations, tracks or trams, whereas the logical network is comprised of concepts and plans, e.g. lines, trips or time tables. Figure 1 shows an extract of an example network.

At the beginning of each *turn*, which is the planned movement of a vehicle through the network on a specific operational day, a tram leaves the *maintenance and storage depot* where it was stored over night. It then travels to the first *platform* of its first *trip*, where the passenger exchange takes place. Platforms are usually unidirectional and always part of a *station*, which combines adjacent platforms under a common name.

After executing the passenger exchange the vehicle travels to the next platform of the trip. The order of platforms which have to be visited is defined by the *line route*. Different line routes can be combined under a common name, thus constituting a *line*. For example Cologne's line 1 (from Junkersdorf to Bensberg and back) actually consists of 27 line routes, 15 of which are east bound and 12 are west bound.

The wagons used by the tram define the maneuvering capabilities and hence how it moves through the network. Table 1 depicts some important characteristics for the three different wagon types which are in use in Cologne's tram network and Figure 2 shows the maneuvering capabilities of wagon type K4000.

The *tracks* between two locations of the network are usually unidirectional, but bidirectional tracks also exist. Some tracks may have speed limitations due to their environment, e.g. inner-city tracks may have a speed limit because of traffic regulations.

While the vehicle travels from one platform to another it may have to traverse *track switches.* These are locations where more than two tracks meet; they can be differentiated between dividing and joining track switches. Like platforms and tracks, track switches are usually unidirectional. All but one of the tracks sharing one side of the track switch must form a curve, which leads to speed limitations that are usually lower than the speed limits on tracks.

The access to track switches (as well as to platforms and track sections) is usually controlled by *traffic lights.* At the end of the operational day the tram travels once again to a maintenance and storage depot. The spatial and chronological order of the vehicles in use on a specific operational day is constituted by the *time table*, i.e. the time table assigns each tram a turn and each turn a set of line routes with starting times.

Characteristics	**K4000**	**K4500**	**K5000**
Length of wagon	29.2 m	29.0 m	29.3 m
Weight of wagon	35.0 t	39.0 t	37.8 t
Maximum velocity	80 kph	80 kph	80 kph
Acceleration	1.3 m/s²	-	1.2 m/s²
Normal brake ability	1.4 m/s²	-	1.2 m/s²
Brake ability for emergency brake	3.0 m/s²	-	2.73 m/s²

Table 1. Characteristics of different wagon types as found in [10], [11] and [12].

2 Modeling Tram Traffic

2.1 Approach

Our approach to model and implement the described system is based on the characteristics of the adopted dynamic-adaptive parallel simulation engine (first described in [6]), which is still under development and was up to now tested on randomized graphs only.

The framework follows a model-based parallelization approach, which tries to exploit the embedded model's intrinsic parallelism. To take maximum advantage of this, the engine is limited to systems that can be considered as sparse, directed graphs, which include many traffic simulation models.

While building the model a number of the applied simulation engine's requirements have to be met. Each model node belongs at every instant to exactly one computational node, which can be a processor or processor core sharing a common cache with its neighbors, or a remote computer connected via a network by message passing. Communication takes place exclusively between computational nodes whose model nodes are connected via edges. The means of communication are transparent to the model nodes. Furthermore the simulation engine takes care of dynamic load balancing, its mechanics are beyond the scope of this paper and are described in [6].

2.2 Physical Network

The tram network is modeled as a directed graph with platforms, tracks and track switches represented by nodes. Every node administrates its currently hosted vehicles. Connections between nodes are represented as edges. Figure 3 depicts an example graph.

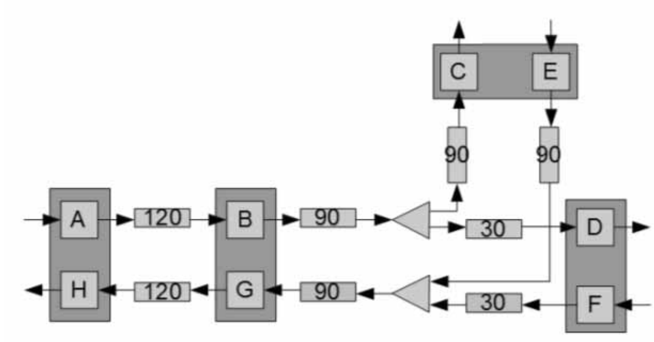


Figure 1. Example graph representing part of a tram network. Squares represent platforms, rectangles tracks and triangles track switches. The darker rectangles around platforms indicates that these platforms form a station.

At any point of time only one vehicle can be located at a platform, which is the main element for modeling boarding and deboarding of passengers. In the real world system passenger exchange is influenced by the platform and day time as well as tram type and passengers (e.g. speed and number). For simplicity's sake we model the boarding and deboarding of passengers as loading time distributions specific to platform and tram type with the combined duration of opening and closing the vehicle doors as minimum value.

Tracks are the only type of node that allows for more than one tram to be located at it at any point of time. The only exceptions to this rule are bidirectional tracks, which have to be exclusively reserved before a vehicle is allowed to enter them. Because the applied ÖPNV data model described in [9] does not allow for bidirectional connections between two locations of the network, they are modeled as two opposed unidirectional tracks. Reservation of one of the coupled tracks then causes blocking of the corresponding opposing track. Tracks also administrate traffic lights located on them.

As in [2] track switches are modeled as transfer points, i.e. they pass trams from an incoming to an outgoing track. Like platforms track switches can only be occupied by one tram at any point of time. Hence they have to be reserved before being entered and unblocked afterwards. Track switches are the only node type that can have more than two neighbors.

As described above, traffic lights are administrated by tracks. Their position at the track is given as an offset related to the beginning of the corresponding track. Phase change is modeled as a function. This is possible because in the described model each traffic light i has constant specific phase lengths t_{red} and t_{green} and subsequently equal cycle lengths

$$t_{cycle} = t_{red} + t_{green}$$

Randomly choosing the time of the first phase change from green to red by

$$t_{init}(i) = -1 * rand(0, t_{cycle}),$$
$$rand(0, t_{cycle}) \sim U(0, t_{cycle} - 1),$$

the current status can be calculated as given in following formula (1):

$$a: \mathbb{N} \rightarrow \{red, green\}$$

$$a_i(t) = \begin{cases} red, if\left(t - t_{init}(i)\right) \bmod t_{cycle} \leq t_{red} \\ green, else \end{cases} \quad (1)$$

Trams must always be located at a node of the network and their main attributes are specified by the type of wagons used. The tram type also holds functions for the maneuvering capabilities.

As an example the velocity during acceleration from zero as a function of time for tram type K4000 is shown in formula (2).

$$v(t) = \begin{cases} 0 & if\ t < 1 \\ \frac{14}{3} * t - \frac{10}{3} & if\ 1 \leq t < 8 \\ 35.33 * \sqrt[3]{t} - 36.66 & if\ 8 \leq t \leq 36 \\ 80 & else \end{cases} \tag{2}$$

Additional tram types can easily be included in the model by extending the abstract base class.

The tram submodel is based upon the event based simulation approach. Thus trams change their state at events of certain types, like stopping, or accelerating, which happen at discrete points of time. As a result of the event handling the system state may change and follow-up events are generated. Those are usually administrated in a priority queue, also called *Future Event List (FEL)*, as described in [1].

During the modeling process fourteen event types were identified (see Table 2).

Trip start	Emergency brake start
Trip end	Acceleration start
Tram standing	Passenger exchange start
Movement start	Track switch reservation
Braking start	Free track switch
Crash	Bidirectional track reservation
Transfer to next node	Free bidirectional track

Table 2. Identified types of simulation events.

As an example Listing 1 shows the handling of event “tram standing” in pseudo code.

2.3 Logical Network

Most parts of the logical network do not have to be modeled explicitly, i.e. a line just combines a set of line routes under a common name and hence can be implemented as a simple string or integer value.

```
Event “tram standing” for tram t do
 if t is located at a stop then
   if passenger exchange completed then
     try to transfer t to next node
         (and if necessary allocate
          following bidirectional
track)
     catch failed transfer by remaining
           to wait for n seconds
   else execute passenger exchange
 else if t is located on a track then
   if t has reached end of track then
     try to transfer t to next node
         (and if necessary allocate
          following switch)
     catch failed transfer by remaining
           to wait for n seconds
   else accelerate
```

Listing 1. Pseudo code algorithm for event type “tram standing”.

A line route on the other hand holds more information and therefore is modeled explicitly. Main component of a line route is a sorted list of identifiers of platforms which have to be visited in this order. Because the ÖPNV data model contains no information about track switch locations on line routes, this information has to be computed prior to the simulation or dynamically before a tram tries to transfer to the next node. In order to identify individual line routes, each one is assigned a name and a unique ID.

Trips allocate a planned starting time to a specific line route and are assigned unique IDs. Each tram then holds a sorted list of trips, which constitutes its turn. The set of turns of a specific operational day constitutes the time table of that day.

2.4 Simulation Infrastructure

In order to meet the requirements of the parallel simulation engine the tram network is divided into disjoint parts, each of which is then allocated to a model node. The special case of assigning the whole network to one model node results in a sequential simulation.

Each model node holds a priority queue of trams located on the part of the network allocated to the node.

When the model node receives the instruction to calculate the next simulation step it first inserts new vehicles, i.e. trams that were sent by neighboring model nodes, into the priority queue. It then instructs each vehicle whose time stamp is equal to the simulation time to execute the next simulation step.

Finally all vehicles that need to be transferred are sent to neighboring model nodes.

3 Simulating Cologne's Tram Network

We apply the developed simulation software to our hometown Cologne's tram network based on the time table data of 2001, as seen in Figure 4. It consists of 528 platforms and 58 track switches connected via 584 tracks. These tracks cover a total length of 407.4 kilometers, resulting in an average track length of 697.6 meters. 15 lines with 182 line routes exist. On each operational day 2,814 trips are executed by 178 trams.

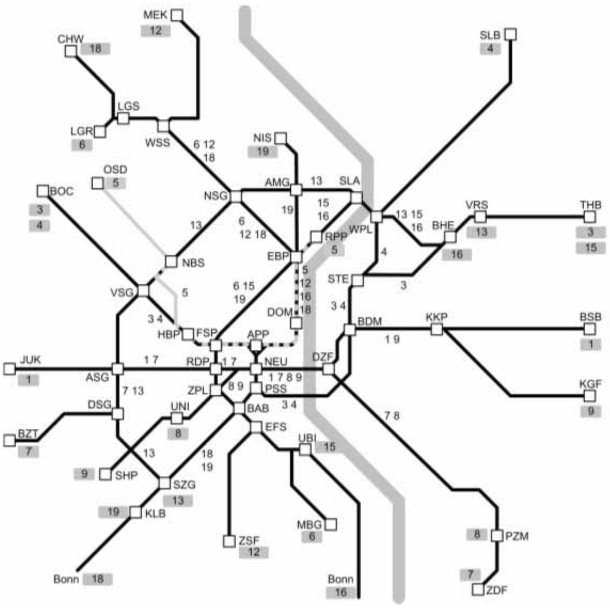


Figure 4. Cologne's tram network in 2001.

We map each node of the graph representing the tram network as a model node and execute 100 simulation runs, yielding an average run time of 348 seconds for a whole operational day.

The results show an average delay of departure over the whole system of 18.67 seconds and a mean delay of 36.05 seconds. During the whole operational day 39,674 departure delays occur, of which 32,389 (81.6%) are less than or equal to 60 seconds (see Figure 5).

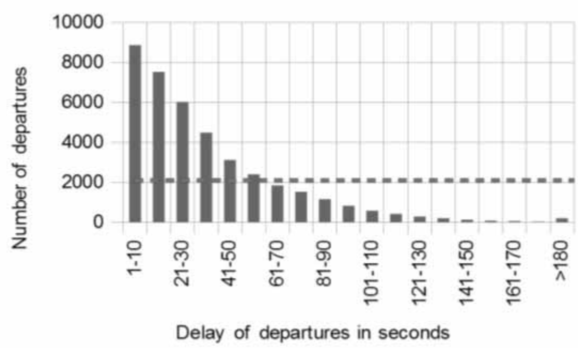


Figure 5. Delay frequencies.

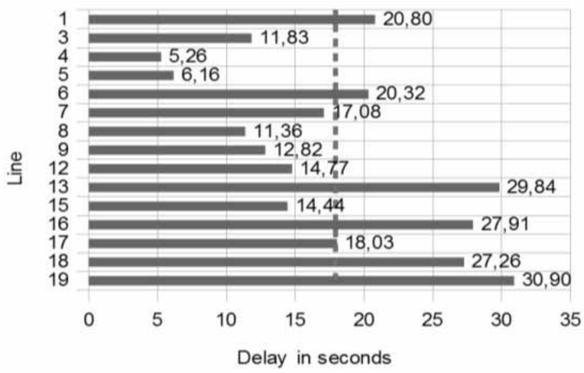


Figure 6. Line delay.

As seen in Figure 6 the lines of the network vary greatly in average delay, mainly due to differences in route length, departure frequencies and inter line dependencies.

For the remainder of this paper we take a closer look at line 5 (see highlighted line in Figure 4) in order to confirm plausibility of our model and to show that the results of our application reflect phenomena observable in Cologne's tram network. Serving 17 platforms line 5 is the shortest line of the network and therefore best qualified for a detailed discussion. About half of the line runs through the inner city, while the other half runs through suburbs. It shares most of its inner city tracks with lines 3, 4, 12, 16, 18 and short parts also with lines 6, 13, 15 and 19. Furthermore for about one third of its tracks line 5 travels underground.

Figure 7 depicts the average delay over the served platforms of trip no. 6 of tram 504, starting at 7:47 at Ossendorf station (OSD) and traveling to Reichenspergerplatz (RPP). During the first half of its trip the tram travels along tracks not shared with other lines.

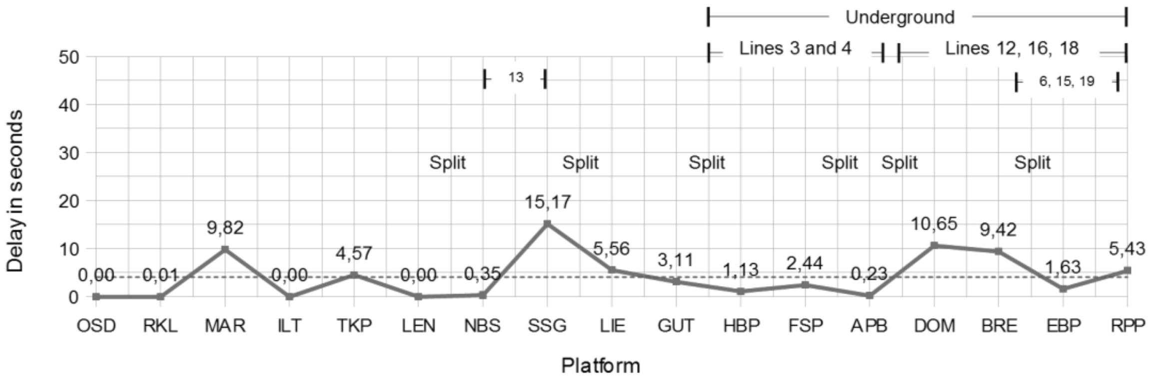


Figure 7. Line 5, Tram 504, Trip 6, starting at 7:47 at Ossendorf.

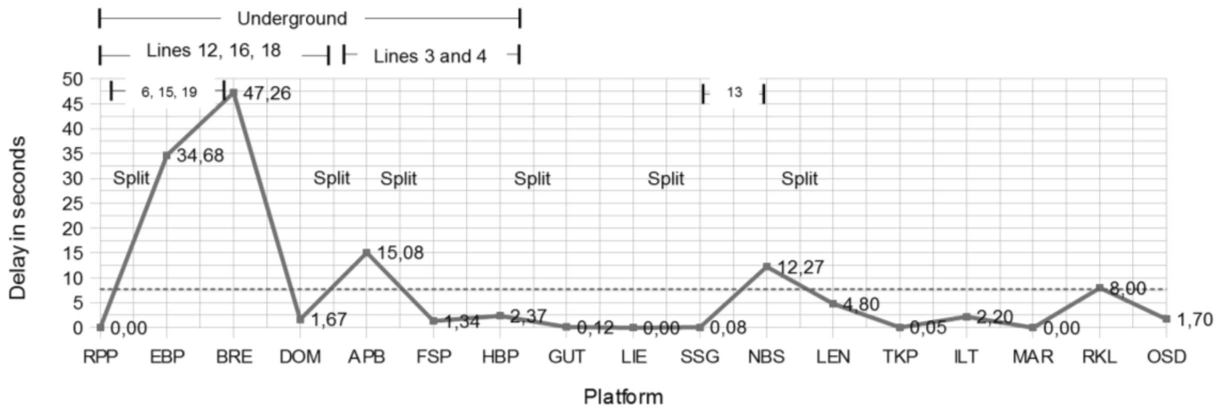


Figure 8. Line 5, Tram 504, Trip 7, starting at 8:21 at Reichenspergerplatz.

The first two peaks in delay at stations Margaretastrasse (MAR) and Takuplatz (TKP) result from a too tight schedule, i.e. the tram needing more than the scheduled 60 seconds to traverse the 700 meter and 580 meter tracks leading to MAR and TKP. On the other hand the planned travel times to the succeeding stations are twice as high, while both tracks are roughly 100 meter shorter. Thus the vehicle is able to eliminate the delay completely.

Though with a length of 280 meter shorter than e.g. the track leading to MAR and having the same planned travel time (60 seconds), a similar effect can be observed between stations Nussbaumerstrasse (NBS) and Subbelratherstrasse/Gürtel (SSG). This is due to the fact that the tram has to pass two traffic lights on the way.

Because traffic lights in the described model have constant phase lengths, the average waiting time t_w at each traffic light can be calculated as seen in following formula (3):

$$t_w = \frac{t_{green}}{t_{cycle}} * 0 + \frac{t_{red}}{t_{cycle}} * \frac{t_{red}}{2} \tag{3}$$

For our experiments we assumed $t_{red} = t_{green} = 30$ seconds, hence from NBS to SSG the tram has to wait 2 * 7.5 = 15 seconds on average, leaving only 45 seconds to traverse the track, coordinate with joining line 13 and exchange pasengers at SSG.

Between SSG and Hans-Böckler-Platz (HBP) the vehicle is able to reduce the delay. The reduction rate flattens after station Liebigstrasse (LIE) because the tram has to pass traffic lights once again. Furthermore after Gutenbergstrasse (GUT) the tram has to coordinate with vehicles of joining lines 3 and 4.

After station Appellhofplatz (APB) lines 3 and 4 separate from line 5 and lines 12, 16 and 18 join. The necessary coordination between the vehicles results in the accumulation of delay at station Dom/Hbf (DOM).

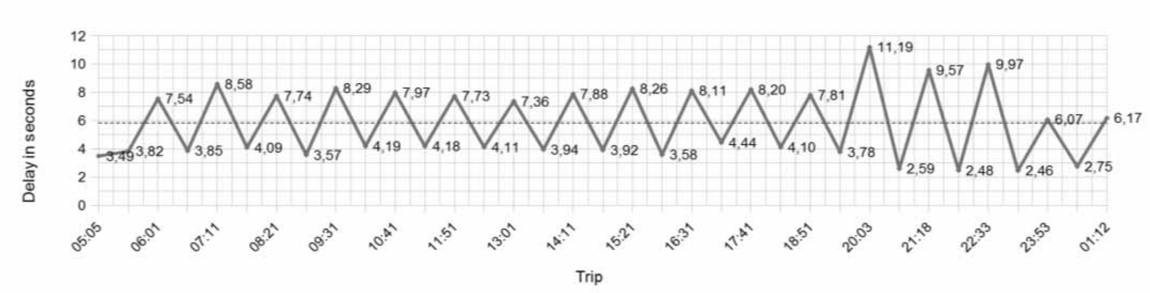


Figure 9. Average delay of trips of tram 504.

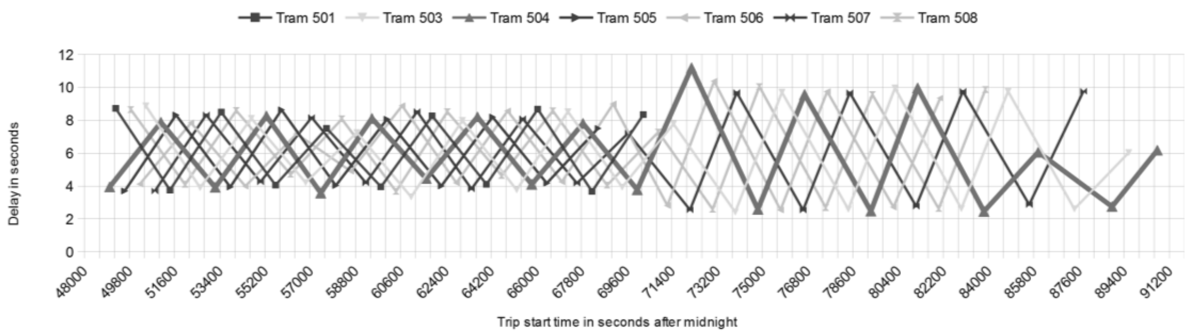


Figure 10. Delay of all trips of line 5 since 13:20.

Figure 8 shows the follow-up trip of tram 504. The increase in delay between RPP and Ebertplatz (EBP) in contrast to the more moderate during the preceding trip can be explained by the significantly smaller safety distance between lines 5 and 18 (one minute compared to three minutes). From Breslauer Platz (BRE) to DOM the vehicle is able to reduce its delay almost completely, while in the opposite direction no such effect can be observed. The cause of this is that the planned travel time from BRE to DOM is 60 seconds higher than the travel time for the opposite direction, accounting for a higher expected time for passenger exchange at Dom/Hbf, which is a major national railway node. Because our model currently does not account for this the simulated vehicle is able to reduce the delay.

Since no vehicle leaves its current platform ahead of the planned departure time no travel time buffer is aggregated, as can be seen between GUT and LIE, where the delay could not be reduced below zero

Observing a vehicle over a whole operational day (tram 504, Figure 9) we see a clear pattern: every trip from RPP to OSD has a higher average delay than trips from OSD to RPP. The only exception to this is the first trip of the operational day which is a short maintenance trip.

The average delay of trips from RPP to OSD is higher than the average delay of trips in the opposite direction, because vehicles traveling from RPP to OSD accumulate a very high delay over the first three platforms where the coordination between lines 5, 6, 12, 15, 16, 18 and 19 is amiss. On the other hand, during trips from OSD to RPP the coordination between vehicles at the critical platforms is considerably better, resulting in a lower average delay.

During the evening hours of the operational day, beginning at 20:00 o'clock, a change in the delay amplitude can be observed (see Figures 9 and 10). The cause of this is twofold. First the tact of the schedule is changed from 10 to 15 minutes in order to reflect lesser demand. Secondly, as a result of the change in tact vehicles are taken out of the system. Thus trams of all lines head for the maintenance and storage depots, which are located at central points in the network, resulting in an increase in utilization of tracks leading to those depots. This worsens the already poor coordination between lines on the outbound tracks. After the second tact change (from 15 to 30 minutes) at roughly 23:00 o'clock coordination between the remaining vehicles gets better again.

Both conditions can be observed for all trams as can be seen in Figure 10, which depicts the average trip delay for all vehicles of line 5 between 13:20 and 01:10.

4 Conclusions and Future Work

In this paper we described our approach for modeling time table based tram traffic. Beginning with a description of the structure of tram networks, which can be considered as a combination of physical and logical components, we described the different entities, e.g. trams, tracks or traffic lights, and their interaction.

After that we characterized our approach for modeling tram networks as graphs with trams as transient entities encapsulating most of the event based simulation logic, using the parallelization framework.

Finally we applied the developed simulation software to Cologne's tram network and analyzed some results. We were able to demonstrate that our application shows the expected behavior and the results reflect the phenomena observable in Cologne's tram network. We also demonstrated real world applicability of the simulation engine.

In further steps the developed model will be applied to other time tables generated with the help of optimization tools as well as real world time tables for further evaluation. First results can be found in the accompanying paper "Simulation and optimization of Cologne's tram schedule" (again, see [7]).

Acknowledgements

This work was partially supported by *Rhein Energie Stiftung Jugend Beruf Wissenschaft* under grant number W-10-2-002. We thank the other members of our work group Patrick Kuckertz and Bert Randerath.

An abbreviated version of this paper was previously published in Bödi, R., Maurer W. (Ed.): Proceedings of ASIM 2011, Winterthur, 2011.

References

[1] J. Banks, J.S. Carson, B.L. Nelson, D.M. Nicol. *Discrete-Event System Simulation*. Upper Saddle River, Pearson, 2010.

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

[3] G. Lückemeyer, E. Speckenmeyer. *Comparing Applicability of Two Simulation Models in Public Transport Simulation.* In M. Becker, H. Szczerbicka (Ed.): ASIM 2006 – 19. Symposium Simulationstechnik. ASIM/Universität Hannover, 2006.

[4] D. Lückerath. *ParSiVaL – Entwurf und Entwicklung einer Anwendung zur parallelen Simulation von schienengebundenem Öffentlichen Personennahverkehr*. Diplomarbeit, Universität zu Köln, 2011.

[5] Oberbürgermeister der Stadt Köln. *Stadtentwicklung in Köln – Mobilitätsentwicklung in Köln bis 2025*. Stadt Köln, 2008.

[6] O. Ullrich. *Eine dynamisch-adaptive Ansatz zur modellbasierten Parallelisierung von Simulationsanwendungen.* Work in progress, Universität zu Köln.

[7] O. Ullrich, D. Lückerath, E. Speckenmeyer. *Simulation and Optimization of Cologne's tram schedule*. SNE Simulation Notes Europe, 22(2), 67-74, ARGESIM Vienna 2012. doi: 10.11128/sne.22.tn.10123

[8] E. Speckenmeyer, N. Li, D. Lückerath, O. Ullrich. *Socio-economic Objectives in Metro Scheduling* (Technical Report). Zentrum für angewandte Informatik Köln, 2011.

[9] Verband Deutscher Verkehrsunternehmen eV. *VDV-Standardschnittstelle Liniennetz/Fahrplan.* VDV-Schriften 452 (2008).

[10] Vossloh Kiepe GmbH. *Elektrische Ausrüstung des Niederflur-Stadtbahnwagens K4000 der Kölner Verkehrs-Betriebe AG*. Druckschrift 00KV7DE, 2003.

[11] Vossloh Kiepe GmbH. *Elektrische Ausrüstung des Niederflur-Stadtbahnwagens K4500 für die Kölner Verkehrs-Betriebe AG.* http://www.vossloh-kiepe.com/vkproduktordner.2008-05-14.1154367607/vkproduktordner.2008-06-30.8585393121/vkproduktordner.2008-05-15.5609169940/vkprodukt.2008-05-14.8640403215/vkprodukt_download accessed on 24.05.2011.

[12] Vossloh Kiepe GmbH. *Elektrische Ausrüstung der Hochflur-Stadtbahnwagen K5000 der Kölner Verkehrs-Betriebe AG.* http://www.vossloh-kiepe.com/vkproduktordner.2008-05-14.1154367607/vkproduktordner.2008-06-30.8585393121/vkproduktordner.2008-05-15.5609169940/vkprodukt.2008-06-04.1250026636/vkprodukt_download accessed on 24.05.2011.

Submitted: September 2011
Revised: July, 2012
Accepted: August 1, 2012

Simulation and Optimization of Cologne's Tram Schedule

Oliver Ullrich[1 *], Sebastian Franz[1], Ewald Speckenmeyer[1], Daniel Lückerath[1,2]

[1]University of Cologne, Inst. f. Informatics, Pohligstrasse 1, 50969 Cologne, Germany;
[2]Cologne University of Applied Science, Cologne, Germany; **ullrich@informatik.uni-koeln.de*

Abstract. In many tram networks multiple lines share tracks and stations, thus requiring robust schedules which prevent inevitable delays from spreading through the network. Feasible schedules also have to fulfill various planning requirements originating from political and economical reasons. In this article we present a tool set designed to generate schedules optimized for robustness, which also satisfy given sets of planning requirements. These tools allow us to compare time tables with respect to their applicability and evaluate them prior to their implementation in the field.

This paper begins with a description of the tool set focusing on optimization and simulation modules. These software utilities are then employed to generate schedules for our hometown Cologne's tram network, and to subsequently compare them for their applicability

Introduction

In many tram networks, several lines share resources like platforms and tracks. This results in very dense schedules, with vehicles leaving platforms every minute at peak times. In order to prevent inevitable local delays from spreading through the network, a feasible schedule has to be robust.

Many additional planning requirements of real world tram schedules originate from political, economical and feasibility reasons. Thus it is not sufficient to exclusively consider general criteria like robustness or operational costs when generating time tables. Typical requirements include fixed start times at certain stations, e.g. interfaces to national railway systems, core lines to relieve high passenger load, e.g. for lines which traverse city centers, warranted connections at certain stations, and safety distances to be complied with at bidirectional tracks. In this paper we present an introduction to our project to generate and evaluate robust time tables which also satisfy given sets of planning requirements. We describe a tool chain which enables us to generate optimized schedules, compare their feasibility and evaluate them prior to application to real world networks.

This paper continues with a description of the current state of the project, focusing on our approaches on optimization and simulation (Section 1). We then present some experimental results obtained by applying the described software to our hometown Cologne's tram network (Section 2). The paper closes with a short summary of lessons learned and some thoughts on further research (Section 3).

1 Simulating and Optimizing Tram Schedules

Our project *'Computer Aided Traffic Scheduling' (CATS)* is built around a database complying with the ÖPNV5 data model released by the *Association of German Transport Companies (Verband Deutscher Verkehrsunternehmen*, see [21]). Visualization, optimization, and simulation modules are connected via operations on the database and through XML configuration files (see Figure 1). Due to its compliance with the ÖPNV5 data model our framework is capable of working on many European tram networks.

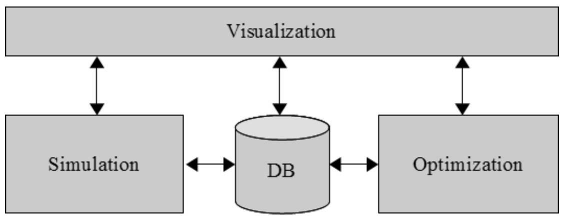


Figure 1. Project modules.

1.1 Optimization of Tram Schedules

Various approaches to optimize tram and railway schedules are known (see e.g. [1, 3, 4, 7, 17, 18, 19]). Most of them aim at one general objective like minimizing vehicle delay (see [17, 19]) or maximizing robustness to restrict the global impact of small, local disturbances (see [4, 7]). Others apply a combination of objectives, like operational profit and robustness in [3], or combining social opportunity cost and operational cost in [18].

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 22(2), 2012, 69-76 | doi: 10.11128/sne.22.tn.10123

Because of the complex nature of the problem, many authors use heuristic approaches like Lagrangian heuristics (see [3]) or simulated annealing (see [18]). Others, like Bampas et al. in [1] introduce exact algorithms for restricted subclasses, like chain and spider networks.

In our project, we combine heuristics and exact methods to generate optimal synchronized time tables for tram networks, targeting maximal robustness and adherence to transport planning requirements at the same time.

We use the scheduled time offset between two consecutive lines departing from a platform as an indicator for robustness. In an assumed tact interval of ten minutes, two lines could be scheduled with equidistant offsets of five minutes, which means that one or both involved vehicles could be late for more than four minutes without consequences for the following tram. Under an extremely unequal split of the available time span into a nine minute offset followed by a one minute offset, the first tram could have a delay of more than eight minutes without consequences to the following vehicle. On the other hand, would the second vehicle be even slightly late, the delay would spread to the follow-up tram. Since we are assuming typically small delays, we see an equidistant distribution as very robust, the occurrence of very small offsets as not robust.

So, to calculate the robustness of a time table λ we examine at each platform h of the network the scheduled time offset $\delta_{f,pred(f)}(\lambda,h)$ between any trip f and its predecessor $pred(f)$, i.e. the time elapsed between the departures of $pred(f)$ and f at the examined platform.

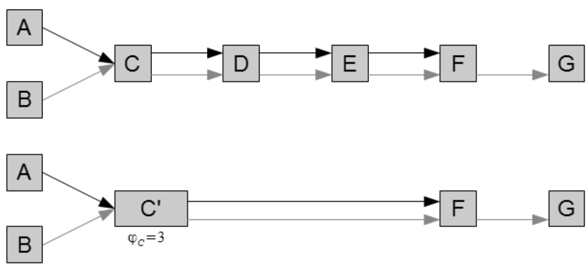


Figure 2. Example of platform reduction.

To reduce complexity we aggregate subsequent similar platforms operated by the same lines to a maximal platform type h', weighted by the number of included platforms φ_h (see Figure 2). The reduced set of platforms is denoted by H'.

To calculate the robustness $\Phi_\mathrm{a}(\lambda)$ of schedule λ, we add the inverse of $\delta_{f,pred(f)}(\lambda,h)$ for all platforms $h \in H'$ and all trips, thus applying a penalty for small safety distances. With F_h representing all trips that serve platform h under schedule λ, the resulting function is as follows:

$$\Phi_\mathrm{a} = \sum_{h\in H'} \sum_{f\in F_h} \frac{1}{\delta_{f,pred(f)}(\lambda,h)} * \varphi_h \tag{1}$$

Given is a set V of planning requirements $v \in V$. In order to calculate the compliance with transport planning requirements we introduce $\rho_v(\lambda) \in \{1,2,3,\infty\}$, the compliance factor of requirement v under a schedule λ. A compliance factor 1 means that the requirement is completely satisfied, 2 and 3 denote tolerable compliance, and ∞ means that the constraint is not met and the time table candidate λ must be rejected. We add the compliance values for all $v \in V$ and get the following:

$$\Phi_\mathrm{b} = \sum_{v\in V} \rho_v(\lambda) \tag{2}$$

Depending on the network under consideration and the number of planning requirements, the two parts of the objective function may not be comparable directly. Thus we define a normalizing factor σ, which reflects the relationship between the theoretically optimal distance $\delta^{opt}_{f,pred(f)}(h)$ and the best possible compliance factor ρ_v^{min}. The theoretically optimal distance $\delta^{opt}_{f,pred(f)}(h)$ of two trips f and $pred(f)$ on platform h is obtained by dividing the tact interval by the number of serving lines at that platform. The best possible compliance factor ρ_v^{min} of a planning requirement v is the minimal value assigned by the planner, independent of the characteristics of the examined solution candidate. Typically $\rho_v^{min} = 1$. We define σ as:

$$\sigma = \frac{\sum_{h\in H'} \sum_{f\in F_h} \frac{1}{\delta^{opt}_{f,pred(f)}(h)} * \varphi_h}{\sum_{v\in V} \rho_v^{min}} \tag{3}$$

Combining $\Phi(\lambda)_a$ and $\Phi(\lambda)_b$ yields the overall objective function $\Phi(\lambda)$ (see formula 4), normalized by σ and weighted by α, the relative weight of the fulfillment of planning requirements.

$$\Phi(\lambda) = (1-\alpha) * \sum_{h\in H'} \sum_{f\in F_h} \frac{1}{\delta_{f,pred(f)}(\lambda,h)} * \varphi_h + \alpha * \sum_{v\in V} \rho_v(\lambda) * \sigma \tag{4}$$

We identify seven types of transport planning constraints: Interval constraints, start time constraints, core line constraints, bidirectional track constraints, turning point constraints, warranted connection constraints and follow-up connection constraints. Upon closer inspection it becomes clear that interval and start time constraints are fundamental and all other constraint types can be expressed using these two. E.g. a bidirectional track constraint can be expressed by two interval constraints covering opposing platforms. Subsequently only interval and start time constraints are considered in the remainder of this paper.

A valid solution also has to adhere to some more restrictions. The first restriction requires each start time μ_i to be inside the tact interval, with $t_{interval}$ being the duration of the interval (see formula 5).

$$\forall \mu_i \in \lambda: \quad 0 \leq \mu_i < t_{interval} \tag{5}$$

Another restriction requires an offset of at least one minute between two departures f and *pred(f)* at each platform $h \in H$ (see formula 6). This means that no platform can be blocked by more than one train at any point of time, the schedule has to be free of collisions.

$$\forall h \in H: \quad \forall f \in F: \quad \delta_{f,pred(f)}(\lambda, h) > 0 \tag{6}$$

To accelerate the computational process the implemented branch-and-bound solver starts with an initial solution computed by a genetic algorithm. The genetic algorithm encodes a time table as one individual, consisting of the first trip start time of each line, i.e. the offset in minutes from the start of the operational day. All other trips follow by the global tact interval. The application generates a start population using random start time values, testing validity against planning constraints and collisions on network nodes.

To reduce computational complexity we apply simple deterministic tournament selection and two-point-crossover (as described in [5]). After evaluation of several mutation methods, including random, minimal, and maximum enhancement mutation we choose a minimal random mutation method that only allows start times to be altered by one minute. We utilize a steady state replacement method, also described in [5]. At the end of each run a hill climbing algorithm is applied to the best individual to further improve its fitness.

As described above we use the best individual encountered by the genetic algorithm to provide the branch-and-bound solver with an initial upper bound, thus avoiding a cold start.

Each inner node of the search tree represents a partial solution of the problem. The root of the tree corresponds to a solution in which no line's start time is fixed. With each level of the tree admissible start times for an additional line are set. For a more detailed discussion of the branch-and-bound method, see e.g. [8].

In order to cut branches off the tree as soon as possible, the objective function of the branch-and-bound algorithm is modified. The set of lines L is divided into subset $\hat{L}$ of lines that are already fixed and subset $\tilde{L}$ of lines that are not yet fixed. Accordingly the set of platforms H is divided into $\hat{H}$ and $\tilde{H}$. $\hat{H}$ includes all platforms which are exclusively served by lines already fixed, while platforms in $\tilde{H}$ are also (or exclusively) served by lines that are not yet fixed. The set of transport planning constraints V is divided into sets $\hat{V}$ and $\tilde{V}$. Set $\hat{V}$ includes all constraints which are dependent on already set lines, correspondingly constraints in $\tilde{V}$ are dependent on lines not yet set. The modified objective function $\Phi'(\lambda)$ is shown below (formulas 7 to 9).

$$\Phi'(\lambda) = (1-\alpha) * \Phi'_a + \alpha * \Phi'_b * \sigma \tag{7}$$

$$\Phi'_a(\lambda) = \left(\begin{array}{l} \displaystyle\sum_{h \in \hat{H}} \sum_{f \in F_h} \frac{1}{\delta_{f,pred(f)}(\lambda, h)} * \varphi_h \\ \displaystyle + \sum_{h \in \tilde{H}} \sum_{f \in F_h} \frac{1}{\tilde{\delta}_{f,pred(f)}(\lambda, h)} * \varphi_h \end{array} \right) \tag{8}$$

$$\Phi'_b(\lambda) = \sum_{v \in \hat{V}} \rho_v(\lambda) + \sum_{v \in \tilde{V}} \rho_v^{min}(\lambda) \tag{9}$$

Here $\tilde{\delta}_{f,pred(f)}(\lambda, h)$ represents the theoretically best safety distance value under consideration of lines already fixed. Again, ρ_v^{min} denotes the optimal compliance factor for constraint v. These values are applied in order to find a lower bound for solution candidates in the current branch of the search tree. For further implementation details, see [6].

1.2 Simulation of Tram Schedules

Most rail-bound traffic simulations are designed for long distance train or railway networks, see e.g. [14, 16]. While those systems feature similarities to tram networks, e.g. passenger exchange or maneuvering capabilities, they differ greatly in important aspects. Tram networks are often mixed, i.e. trams travel on underground tracks as well as on street level, and are thus subject to individual traffic and corresponding traffic regulation strategies.

Subsequently, tram behavior is a mixture between train and car behavior, e.g. line-of-sight operating / driving. Therefore a simple adaption of railway simulation methodologies is not feasible.

Bearing the similarities with individual traffic in mind Joisten (see [9]) implemented an adapted Nagel/ Schreckenberg model (see [15]) for tram simulation, which suffered from the setbacks of the high aggregation inherent to cellular automatons (see [11]). Therefore Lückemeyer developed an event based simulation model which avoids some of those setbacks as described in [10, 11]. To further eliminate inaccuracies we apply an updated model, is described in detail in the accompanying article 'Modeling time table based tram traffic' ([13]).

Our application is based upon a model-based parallelization framework (described in [20]), which exploits the embedded model's intrinsic parallelism. The mixed tram network is modeled as a directed graph with platforms, tracks and track switches represented by nodes. Connections between nodes are represented as edges. Figure 3 shows part of an example network, which is mapped on the graph depicted in Figure 4, where squares represent platforms, rectangles tracks and triangles track switches. The dark rectangles around platforms indicate that these platforms form a station.

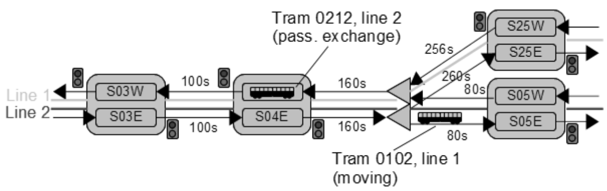


Figure 3. Part of a tram network.

The distributions for the duration of passenger exchange are specific to platform and tram type with the combined duration of opening and closing the vehicle doors as minimum value.

Vehicles encapsulate most of the simulation dynamics, which are based upon the event based simulation approach (as described in [2]). Thus trams change their state at events of certain types, like stopping or accelerating, which happen at discrete points in time. These state changes may trigger a change in the overall system state and generate follow-up events, which are administrated in a priority queue.

Main tram attributes are specified by the type of tram, which holds functions for the maneuvering capabilities, e.g. acceleration and braking.

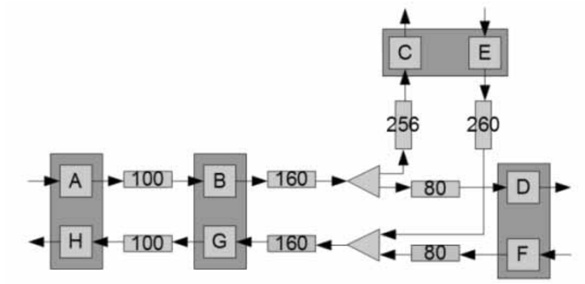


Figure 4. Example graph.

2 Optimizing Cologne's Tram Network

We apply the developed software suite to our hometown Cologne's tram network based on the time table data of 2001 (see figure 5). It consists of 528 platforms and 58 track switches connected via 584 tracks, which cover a total length of 407.4 kilometers. 15 lines with 182 line routes are served by 178 vehicles which execute 2,814 trips per operational day.

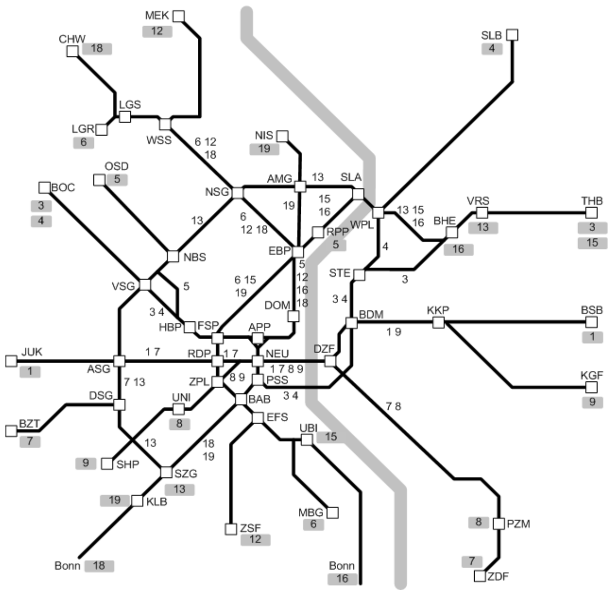


Figure 5. Cologne's tram network in 2001.

For optimization purposes, we only consider the 36 major routes on peak hours. The remaining 146 minor routes are usually trips between the start or end point of a regular trip and depots, or other maintenance trips at the rim of the network. For the following optimization run, we assume a tact interval of ten minutes, and define a set of example constraints, which can be decomposed to two start time constraints and 37 interval constraints.

These include minimum turn-around times at line ends, an additional core line 1A to satisfy high demand for line 1 in Cologne's town center, guaranteed connections between certain lines, and fixed start times at the Bonn national railway hub. A more detailed description of the conducted experiments can be found in [20].

2.1 Comparing Tram Schedules

From the genetic algorithm's initial pool of valid solution candidates we randomly pick a schedule A with an objective function value of 214.714 (see Table 1). After a 8.5 hours run, the optimizer yields a pool of 60 best solutions with objective function values of 180.696, from which schedule B (again, see Table 1) is randomly selected.

To begin with a more general view, we pick ten more schedules each out of both solution pools and execute ten simulations runs for each of those 20 schedules. The runs under the initial schedules yield an average delay of departures of 18.9 seconds. Under the best schedules the average delay is 15.4 seconds, which means a reduction of 18.6 percent or 3.5 seconds.
The frequency distribution of occurring delays was also collected (see Figure 6).

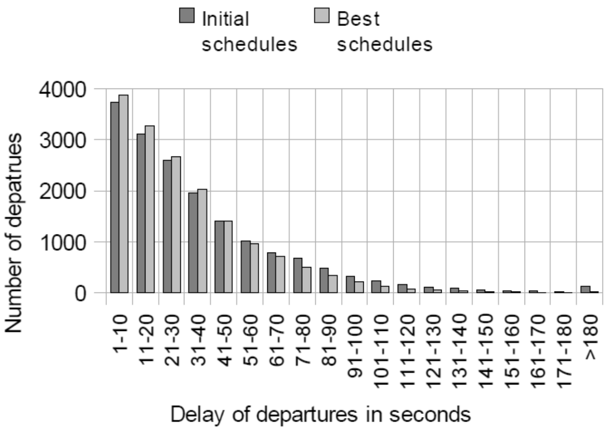


Figure 6. Frequency of all delays.

While the optimal schedules yield more small delays (up to 60 seconds) than the initial schedules, they yield a lower number of bigger delays of more than 60 seconds (see Figure 7): The random schedules result in an average number of 3,095.6 departures with a delay of more than 60 seconds, under the best schedules this number is down by 987.6 departures or 31.9 percent to 2,108.0 departures. The total number of delayed departures is reduced from 16,923.6 under the random schedules by 602.0 departures or 3.6 percent down to 16,321.6.

A higher robustness can help to reduce the number of bigger delays by preventing small delays from accumulating over the simulation run. Under optimal schedules with their better distributed time offsets, the inevitable small delays stay small, so their number is higher than under random schedules. On the other hand, bigger delays cannot build up under robust schedules, so their number is smaller than under random schedules.

Furthermore, we examine both schedules A and B by executing 100 simulation runs each and comparing the results. Schedule A yields an average line delay of 16.5 seconds, which gets reduced under schedule B by 16.4% or 2.7 seconds to 13.8 seconds.

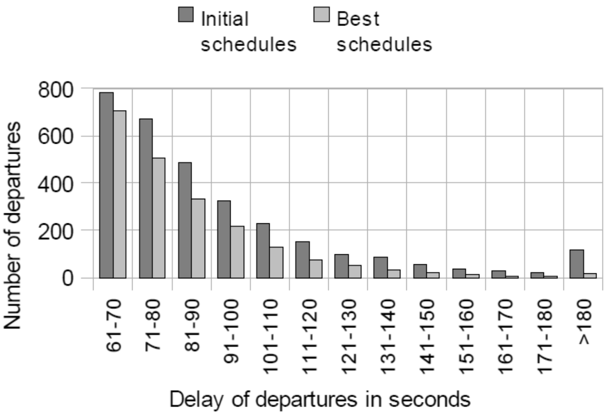


Figure 7. Frequency of delays higher than 60 seconds.

As seen in Figure 8 and Table 2, implementation of schedule B enhances punctuality of most lines at least marginally. Especially the improved punctuality of lines 13 and 18 (32 and 25 percent respectively) show the better coordination between lines under schedule B. Under schedule A, the northbound route of line 13 stands out in coordinating especially bad with line 7: Although the joining line 7 is scheduled to serve a row of platforms beginning with Dürener Straße/Gürtel (DSG, again see Figure 5) two minutes after line 13, its vehicles often reach the first common track switch ahead of schedule, thus blocking it for the already late trams of line 13. Before entering each of the following common stations, these trams have to wait for line 7 to clear the platforms, thus instantly getting a delay of at least 120 seconds. Only after the end of the shared area at Aachener Straße/Gürtel (ASG), the vehicles can begin to regain part of their lost punctuality.
At the other end of the spectrum, Lines 3, 4, and 5 do not improve on their comparatively low delay, or even yield a slightly higher average delay than under schedule A.

All three lines are laid out comparatively well in the random schedule A. The westward branch of line 5 is scheduled with an exceptionally high clearance in some areas of the town center, rendering it insusceptible to delays of preceding trains (see Figure 9).

Line	A – Initial schedule		B – Best schedule	
	Forward	**Backward**	**Forward**	**Backward**
1	1	7	7	6
1A	6	4	1	0
3	3	3	4	0
4	7	0	6	7
5	5	2	9	6
6	1	7	7	6
7	0	9	7	0
8	6	7	1	7
9	5	4	3	7
12	6	7	4	2
13	7	6	0	4
15	4	6	6	9
16	6	0	3	5
18	9	7	6	5
19	9	0	7	9

Table 1. Comparing schedules: Lines.

A closer view on the behavior of line 5 (though under a different schedule) is presented in the accompanying paper 'Modeling time table based tram traffic' (see [13]).

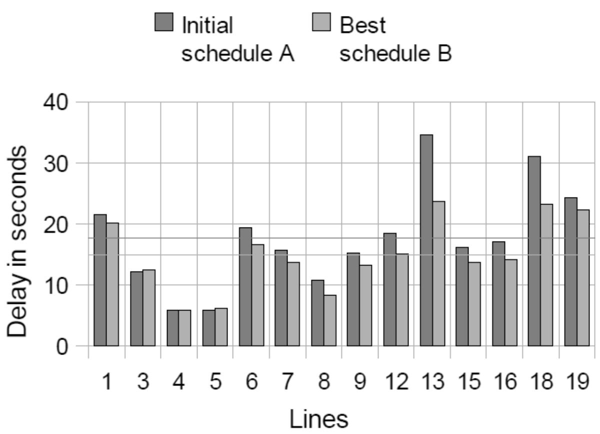


Figure 8. Average delay of lines.

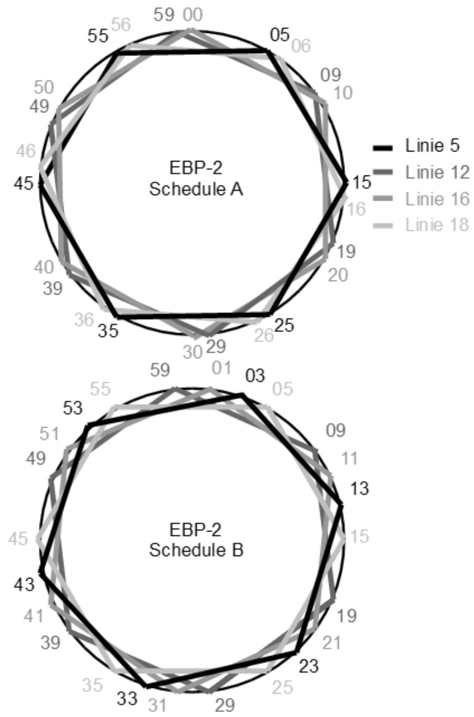


Figure 9. Scheduled departures at platform EPB-2 under schedule A and schedule B.

Simulation data collected at the important hubs Barbarossaplatz (BAB-1 to BAB-4), Ebertplatz (EBP-1 to EBP-4), and Neumarkt (NEU-1 to NEU-4) is presented in Figure 10 and Table 3. Under schedule B, delay was reduced significantly at each of those platforms, on average by 3.7 seconds or 17.6 percent. The increase in punctuality can be explained by the better reliability of the frequenting lines under the optimized schedule.
The high base levels of delay at some platforms (like NEU-2, EBP-4, and BAB-1) is obviously independent of the applied schedule and has therefore to be based on the properties of the surrounding parts of the network.

For instance, the track leading up to NEU-2 has a planned travel time of 120 seconds, including passenger exchange on the platform. It is 880 meters long and because of crossing streets and pedestrian crossings divided by six traffic lights. Because of the applied global phase length of 30 seconds a tram has to wait at each of these lights for 7.5 seconds on average, accumulating to 45 seconds of standing time. Only 75 seconds of scheduled time remain for the actual traversing of the track, including accelerating and braking in front of traffic lights three times on average, and the passenger exchange. This is obviously not enough, resulting in the observed base delay.

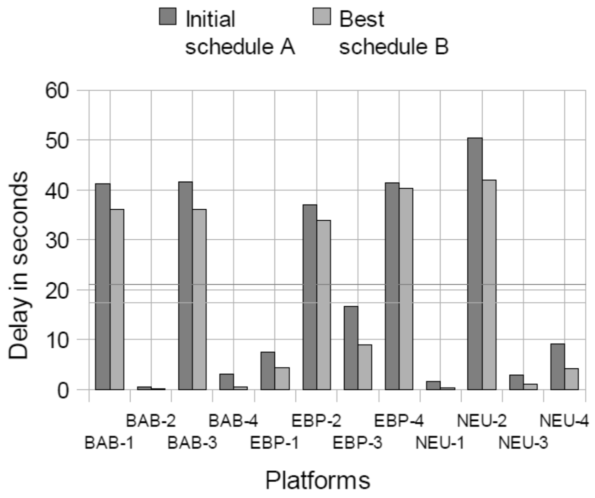


Figure 10. Average delay at platforms.

Line	Average Delay		Abs. gain	Rel. gain
	A	**B**		
1	21.5	20.2	1.4	0.06
3	12.2	12.4	-0.2	-0.01
4	5.9	5.9	0.0	0.00
5	5.8	6.2	-0.4	-0.06
6	19.3	16.6	2.7	0.14
7	15.7	13.7	2.0	0.13
8	10.8	8.3	2.5	0.23
9	15.2	13.3	1.9	0.13
12	18.4	15.0	3.4	0.19
13	34.6	23.6	11.0	0.32
15	16.1	13.7	2.4	0.15
16	17.0	14.1	2.9	0.17
18	31.1	23.2	7.9	0.25
19	24.3	22.2	2.1	0.09
Average	16.5	13.8	2.7	0.16

Table 2. Comparing schedules: Lines.

A similar situation can be found at other platforms like EBP-4 or BAB-3. The relatively low planned travel times for the up-leading tracks correspond to long tracks with several traffic lights, switches and/or other lines that have to be maneuvered. Thus, a base delay is inevitable.

Other platforms have a lower or almost no base delay: i.e. the platform NEU-1 is preceded by a track with a length of 590 meters with a planned travel time of 180 seconds. The vehicles have to wait at two traffic lights, which leaves enough time to arrive at the platform without delay.

Platform	Average Delay		Abs. Gain	Rel. Gain
	A	**B**		
BAB-1	41.3	36.1	5.2	0.13
BAB-2	0.5	0.1	0.3	0.72
BAB-3	41.6	36.1	5.6	0.13
BAB-4	3.0	0.5	2.5	0.83
EBP-1	7.5	4.4	3.0	0.41
EBP-2	36.9	33.9	3.0	0.08
EBP-3	16.6	8.9	7.7	0.46
EBP-4	41.4	40.4	1.0	0.02
NEU-1	1.5	0.3	1.1	0.77
NEU-2	50.3	41.9	8.3	0.17
NEU-3	2.8	1.1	1.7	0.61
NEU-4	9.1	4.2	5.0	0.54
Average	21.0	17.4	3.7	0.18

Table 3. Comparing schedules: Platforms.

3 Conclusion and Future Work

In this paper we presented a tool chain to generate and evaluate tram schedules. The described optimization module is capable of generating robust time tables which fulfill planning requirements of real world projects. We also presented a simulation engine which makes it possible to test real and generated schedules for their applicability and so to further validate them.

We applied the described tool chain to our hometown Cologne's mixed tram network. A random but valid time table A was compared to a resulting best schedule B. As to be expected, the average delay under schedule B is significantly lower than that under schedule A. Most lines and all of the examined core platforms gain punctuality. At least part of the remainig delay can be explained by properties of the underlying network.

In further steps more detailed studies of tram networks and schedules will be carried out, including Cologne's new underground tracks currently under construction, which are designed to relieve the central Neumarkt tunnel. We found it desirable to be able to manually apply small incremental changes to a schedule while getting instant visual assessment of expected consequences. A tool with those capabilities is in the planning stage. Furthermore the optimizer module will be parallelized to further reduce its run time.

Especially the applied branch-and-bound algorithm's load can be balanced relatively easy, so the application should scale well.

Acknowledgments

This work was partially supported by *Rhein Energie Stiftung Jugend Beruf Wissenschaft* under grant number W-10-2-002. We thank the other members of our work group Patrick Kuckertz and Bert Randerath.

An abbreviated version of this paper was previously published in Liu-Henke, X., Buchta, R., Quantmeyer, F. (Ed.): Proceedings of ASIM-Workshop STS/ GMMS 2012. ARGESIM/ASIM Pub. TU Vienna/ Austria, Wolfenbüttel, Feb. 23-24, 2012, 279-289.

References

[1] E. Bampas, G. Kaouri, M. Lampis, A. Pagourtzis. *Periodic Metro Scheduling*. In: Proceedings of ATMOS, 2006

[2] J. Banks, J. S. Carson, B.L. Nelson, D. M. Nicol. *Discrete-Event System Simulation.* Pearson, 2010.

[3] V. Cacchiana, A. Caprara, M. Fischetti. *A Lagrangian Heuristic for Robustness, with an Application to Train Timetabling.* Transportation Science, to appear.

[4] G. Caimi, M. Fuchsberger, M. Laumanns, K. Schüpbach. *Periodic Railway Timetabling with Event Flexibility*. In: Networks, 2010, Volume 57, Number 1, pp. 3-18.

[5] J. Dréo, A. Pétrowski, P. Siarry, E. Taillard. *Metaheuristics for Hard Optimization.* Springer, 2006.

[6] S. Franz. *Entwurf und Entwicklung eines mehrstufigen Optimierungsverfahrens für Stadtbahnfahrpläne unter Berücksichtigung verkehrsplanerischen Vorgaben*. Diplomarbeit, Univ. Köln, 2011.

[7] Z. Genç. *Ein neuer Ansatz zur Fahrplanoptimierung im ÖPNV: Maximierung von zeitlichen Sicherheitabständen*. Dissertation, Mathematisch-Naturwissenschaftliche Fakultät, Universität zu Köln, 2003.

[8] T. C. Hu. *Combinatorial Algorithms*, Addison Wesley, 1982

[9] M. Joisten. *Simulation von Fahrplänen für den ÖPNV mittels Zellularautomaten.* Diplomarbeit, Univ. Köln, 2002.

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

[11] G. Lückemeyer, E. Speckenmeyer. *Comparing Applicability of Two Simulation Models in Public Transport Simulation.* In: Becker, M., Szczerbicka, H. (Ed.): Proceedings of ASIM 2006, Hannover, 2006.

[12] D. Lückerath. *Entwurf und Entwicklung einer Anwendung zur parallelen Simulation von schienengebundenem Öffentlichen Personennahverkehr*. Diplomarbeit, Univ. Köln, 2011.

[13] D. Lückerath, O. Ullrich, E. Speckenmeyer. *Modeling Time Table based Tram Traffic.* SNE Simulation Notes Europe, 22(2), 59-66, ARGESIM Vienna 2012. doi: 10.11128/sne.22.tn.10121

[14] D. Middelkoop, M. Bouwman. *SIMONE: Large Scale Train Network Simulations*. In: B.A. Peters, J.S. Smith, D.J. Medeiros, M.W. Rohrer (Ed.): Proceedings of the 2011 Winter Simulation Conference, Arlington, 2001.

[15] K. Nagel, M. Schreckenberg. *A cellular automaton model for freeway traffic*. Journal de Physique I, Volume 2, Issue 12, December 1992, pp. 2221-2229.

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

[17] A. Schöbel. *A Model for the Delay Management Problem based on Mixed-Integer-Programming*. In: Proceedings of ATMOS, 2001.

[18] E. Speckenmeyer, N. Li, D. Lückerath, O. Ullrich. *Socio-Economic Objectives in Tram Scheduling*. Technical Report, Universität zu Köln, to appear.

[19] L. Suhl, T. Mellouli. *Managing and preventing delays in railway traffic by simulation and optimization*. In: Mathematical Methods on Optimization in Transportation Systems 2001, pp. 3–16.

[20] O. Ullrich. *Ein dynamisch-adaptiver Ansatz zur modellbasierten Parallelisierung von Simulationsanwendungen.* Mathematisch-Natur-wissenschaftliche Fakultät, Univ. Köln, work in progress.

[21] Verband Deutscher Verkehrsunternehmen e.V. *VDV-Standardschnittstelle Liniennetz/Fahrplan,* VDV-Schriften 452, 2008

Submitted: March 2012
Revised: June, 2012
Accepted: August 1, 2012

A Simulation-based Approach on Robust Airline Job Pairing

Patrick Kuckertz[1,2 *], Oliver Ullrich[1], Hubert Randerath[2]

[1]University of Cologne, Inst. f. Informatics, Pohligstrasse 1, 50969 Cologne, Germany;
[2]Cologne University of Applied Science, Cologne, Germany; **kuckertz@informatik.uni-koeln.de*

Abstract. Job paring, i.e. the composition of duty rosters from single activities, is an important part of the airline operations planning process. With labor costs being a major factor in an airline's cost structure, such personnel schedules have to ensure efficiency to be of practical relevance. At the same time they have to improve customer acceptance by offering best possible robustness, keeping inevitable local delays from spreading through the airline's flight network.

In this paper we present a project currently in development which aims for generating robust personnel schedules for airline operations. The resulting tool set will allow us to effectively allocate flight personnel, using optimization and simulation techniques to generate and compare schedules with respect to their applicability and their demand for standby personnel, and to evaluate them prior to their implementation in the field.

Introduction

During their extensive process of operations planning airlines are challenged by a set of interdependent planning problems (see Figure 1). This process starts with the design of the flight schedule and the assignment of aircraft types to the flights. It continues with the routing of individual aircrafts and the determination of crew schedules, and is concluded by short-term flight plan management and recovery measures.

Within this process the construction of a valid and efficient operations schedule for flight personnel is one of the most complex tasks. A part of this task is the crew pairing procedure which is concerned with the construction and optimal combination of anonymous crew rotations in order to cover all flights of a given flight schedule while complying with a multitude of regulations coming from labor legislation (see [4]), union agreements and operational procedures.

The majority of existing studies analyzes the *crew pairing problem* (CPP) against a cost reducing background due to its high economic significance (see e.g. [2], [8]). The use of costs as exclusive quality objective however may lead to personnel schedules with a low degree of fault tolerance and a high degree of delay propagation. In order to confine occurring disruptions and to support practical applicability a personnel schedule has to be robust.

This paper describes and outlines a project in development which aims for a better understanding of robust personnel schedules. The project follows a more detailed approach than the CPP describes by not dividing tasks on crew level but on the level of individual crew members, leading to a *job pairing problem* (JPP). In the context of robustness this approach is more realistic, since delays and drop outs of individuals can be accounted for.

Furthermore individual qualifications can be incorporated which enables the analysis of efficient substitution strategies and standby structures. This approach also allows a more detailed view on the fault propagation in personnel employment strategies. Schedules resulting from our optimization process are to be simulated under realistic conditions. A concurrence of results of a robustness assessment by a static objective function with those of a dynamic simulation would demonstrate a certain suitability of practical use of our approach.

The remainder of this paper is organized as follows. In Section 1 our project is introduced. Four subsections explain technical backgrounds and give insights into different project modules with their objectives and approaches. Section 2 concludes with a summary and some thoughts on future work.

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 22(2), 2012, 77-82 | doi: 10.11128/sne.22.tn.10125

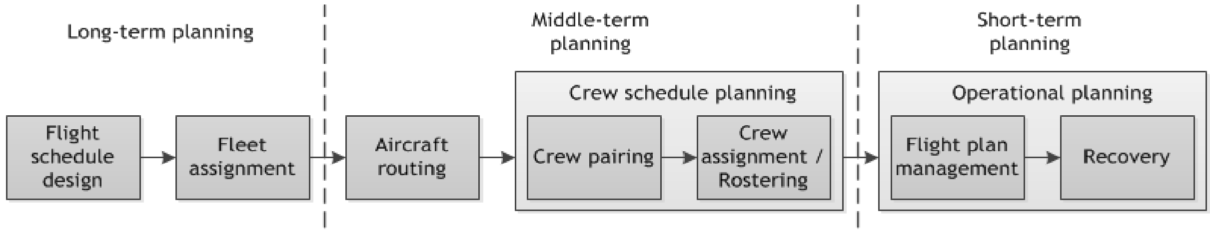


Figure 1. Operations planning process of airlines.

1 Project Approach

Our project *Dynamic Optimization of Group Schedules* (DOGS) is build around a database containing airline schedule and network data. A network generator, simulation, optimization, and evaluation modules are connected via operations on the database and through XML configuration files (see Figure 2).

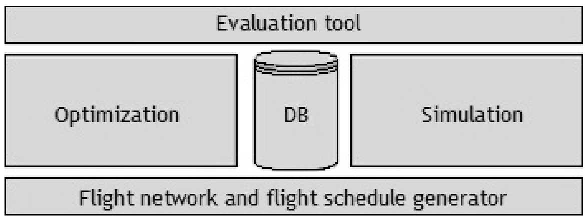


Figure 2. Modular project architecture.

1.1 Technical Background

The development of aircraft rotations is preceded by flight schedule design and fleet assignment which are both based upon passenger demand forecasts (see figure 1). Results of these planning steps are an airline's flight connections as well as the allocation of aircraft types to these connections. *Flight connections* are defined by their origin and destination airports as well as by their departure and arrival times. Aircraft types differ e.g. in passenger capacities and personnel requirements. All this information merges into the flight schedule which serves as input to the crew scheduling process. During the job pairing, which is part of crew scheduling, tasks are combined and packaged. In the following crew assignment or rostering phase these work packages are assigned to members of the flight personnel.

A *flight schedule* provides detailed information about flight connections, informing about the time intervals and weekdays a connection is carried out. Connections in a schedule are identified by flight numbers while individual flights are identified by their connections and days of departure.

There are different types of connections to be found in a schedule, depending on their number of *flightlegs*. Figure 3 shows a diagram of a flight schedule fragment in which flightlegs are pictured as arrows. A non-stop connection, also called *non-stop flight*, has no stops between its airports of origin and destination and therefore only one flightleg. A direct connection, also called *direct flight*, has at least one intermediate stop and thus consists of two or more flightlegs. It does not include any changes of aircraft and its flightlegs operate under a single flight number. Examples can be found in Figure 3, connecting the airports A and C. A non-stop flight between these airports consists only of flightleg L4, while a direct flight stopping at airport B consists of flightlegs L1 and L2. Within a schedule the type of a connection is denoted by the number of stops it includes.

For the JPP not all connections found in a flight schedule are considered. To avoid redundant information direct flights are ignored since they are composed of non-stop flights already named in the schedule. An airline schedule often contains connections actually carried out by alliance partners. This way a flight might be offered by different airlines under more than one flight number, allowing customers to book at their preferred airline in their own language and currency. Those *code share flights* have to be disregarded since we only want to solve the JPP for single airlines.

The connections an airline offers form its *flight network* which can be viewed as a graph with airports being nodes and flight connections being directed edges. Flight networks of large airlines often show hub and spoke structures which support efficient operations (see [7]). Coordinated with adequate schedules they provide passengers with a manifold choice of connections and short waiting times. Commonly airlines choose large airports with strategically favorable positions within their networks to serve as *hubs*. Hubs are usually fully interconnected. *Spokes* connect the hubs to all other airports which are accessed by the airline.

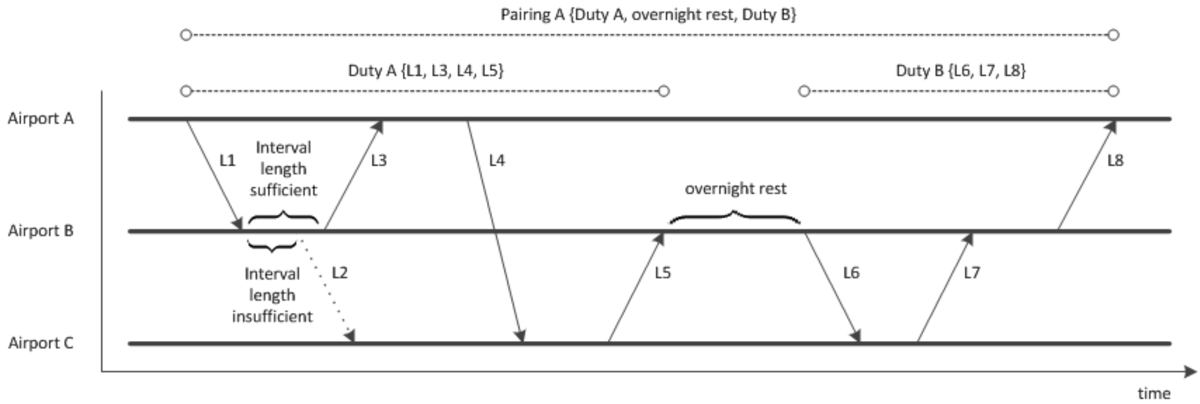


Figure 3. Diagram of a flight schedule fragment.

Within such a network structure the surrounding airports are normally not interconnected with the possible exception of *shuttle connections* extending spokes to other smaller airports which have no connections to a hub themselves.

Also depending on their relative position within the network airlines *choose* at least one airport to serve as *crew base*. An airport is called a crew base if it is the place of employment of airline's personnel. Another term used by airline personnel is *home base* which is the employees' view on a crew base. Each employee has exactly one home base while a crew base must be home base to at least one employee. Figure 4 illustrates the described network structure including crew bases.

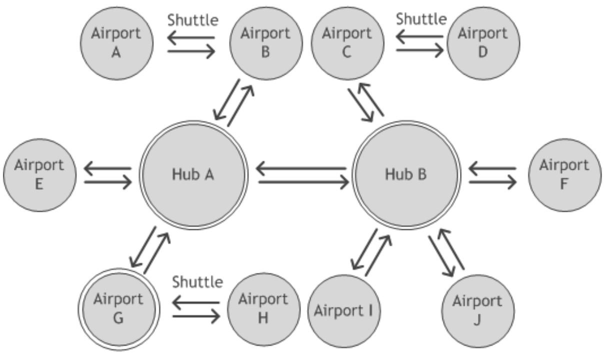


Figure 4. Example of an airline hub and spoke network structure (double lined circles picture crew bases).

1.2 Modeling

Following our job pairing approach, each flightleg brings up a number of *jobs*, i.e. single tasks, all requiring individual combinations of professions and qualification profiles. Depending on aircraft type, number of passengers and flight distance, different sizes of flight deck and cabin crews are mandatory. Different aircraft types and countries of origin and destination require different piloting, language and service skills.

During the job pairing the jobs of all flightlegs have to be assorted into work packages which will be assigned to flight personnel in the following rostering process. Jobs are bundled into *duties* which can be viewed as single workdays. The work packages, called *pairings*, again are bundles of duties with overnight rest periods in between (see Figure 3). They are round trips, starting and ending at the same crew base. The allowed numbers of take-offs and landings within duties and pairings, maximum flying and service times, minimum rest periods and other work rules concerning the packaging process are determined by public authorities and are further subject to operational procedures and union agreements. During the pairing process it may become necessary to reallocate flight personnel to other airports. The transportation of off duty personnel is called *deadhead*.

The current state of our project's data model is pictured in Figure 5. The entity relationship diagram (see [6]) illustrates the composition and relationships of the entities substantial for an airline operations planning process. The structure of the project's database is derived from this diagram.

The scope of the project includes the development of a flight network and flight schedule generator (see Figure 2). With this tool a set of realistic and hypothetical test instances are to be generated to support the robustness analysis. Assessing diverse instances may yield information about the underlying graph structures' influences on the robustness potential. The network graphs of past flight schedules undergo a structural analysis regarding connectivity, reachability and distance measures.

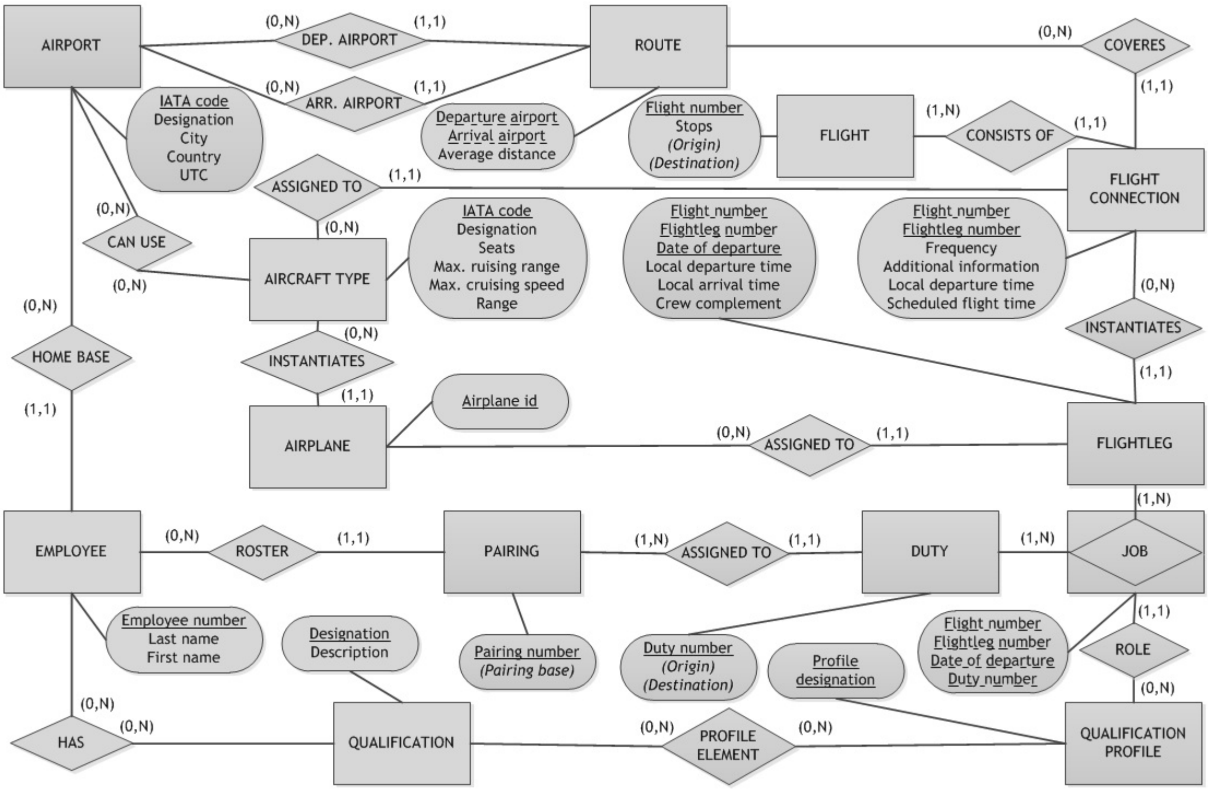


Figure 5. Entity relationship diagram of fundamental data elements within an airline operations planning process.

Once adequate parameters and realistic specifications have been found the algorithm's method of operation has to be determined. After applying a few modifications the R-MAT generator described in [3] might be a promising candidate.

1.3 Optimization

The JPP is a large scheduling problem whose complexity grows with each additional variable representing jobs, qualification requirements and types of work shifts. Due to its combinatorial structure the number of possible solutions is huge. Problem instances with over 1,000 flights a day and a monthly coordination of over 15,000 crew members are not uncommon. In addition a wide spectrum of government regulations upholding aviation safety has to be respected (see [4]).

Cost reduction is the traditional motivation of research on this topic. Personnel costs account for the second highest part of an airline's overall expenses, right after fuel costs which hardly can be impaired (see [7]). The primary aim of the project presented in this paper however is not to reduce the costs of a flight schedule but to improve its robustness.

A robust schedule is to be distinguished by a low rate of delay propagation and a high fault tolerance. Delays and drop outs of personnel members or flightlegs cannot be fully avoided, but measures can be taken to reduce their occurrence probabilities and possible consequences for the flight schedule.

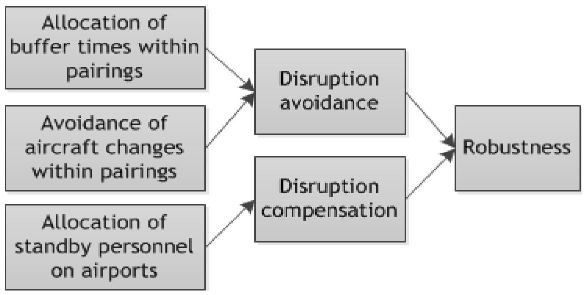


Figure 6. Overview over robustness improving measures during crew pairing.

Each step of the airline operations planning process has its own options to account for disruptions. Crew pairing provides measures to avoid disruptions as well as to compensate for them (see Figure 6). Our project's optimization approach focuses on disruption avoidance.

One policy to create flight schedules with a maximum of stability is to demand a minimum time interval between two consecutive jobs to buffer delays. Figure 3 illustrates that a job of flightleg L2 cannot follow a job of L1 within one personnel member's duty because of the insufficient length of the intermediate time interval. Another measure is the minimization of the number of personnel's aircraft changes within a pairing, reducing dependencies between aircraft rotations and hence delay propagation.

The CPP is often discussed in literature, and a plurality of mathematical models and solution approaches are presented. For our project we haven't yet decided which approaches to adjust to our JPP. Thus we describe a common crew pairing procedure at this point. Crew pairing divides all flightlegs of a given flight schedule into pairings. The problem of covering each flightleg exactly once by a single pairing is described by the *set partitioning problem* (see [2]). In order to include deadheads into the process of optimization the coverage of a flightleg by more than one crew, and hence more than one pairing, must be allowed. The formulation as a *set covering problem* includes the condition to cover each flightleg at least once (see [7]). Solving the CPP for a major airline includes a large set of pairings which leads to a huge number of possible combinations.

Because of its large scale the CPP is often divided into a master problem and a subproblem. The subproblem, including only a manageable amount of pairings, is solved and then iteratively expanded by column generation. Applying the local search heuristic 2-opt (described in [8]), the size of the subproblem stays constant because promising new pairings replace pairings of the previous solution. A common approach for approximating a global optimum is described by the *restricted shortest path problem* (see [9]). Here a problem's graph structure is used to evaluate the quality of all pairings outside the current subproblem so that only the most promising pairings have to be calculated in the next iteration.

Commonly these procedures are used to optimize a cost function. The costs of a pairing can be determined by measuring its time consumption. Gopalakrishnan et. al. define the costs by the difference between the *time away from base* and the *flying time* (see [7]). The time away from base is the time interval between leaving and returning to a crew base.

The flying time is the summation of the differences between the arrival and departure times of all the pairing's legs. This calculation determines non-productive waiting times of pairings. Analogous to [5] we want to treat the aspect of robustness using penalty costs for insufficient intervals between flightlegs and for aircraft rotation changes. The formulation of the robustness objective as a cost reduction problem allows the use of already approved optimization procedures.

1.4 Simulation

We plan to develop a model and implement an application to simulate flight schedules. This will enable us to evaluate given personnel schedules prior to their implementation in the field and to compare schedules generated by optimization methods with respect to their applicability.

Schedules considered feasible by a static objective function, can be evaluated for their dynamic applicability, and thus lead to a higher degree of validity.

Another focus of the simulation system lies on disruption compensation, i.e. to evaluate a given personnel schedule for its recoverability characteristics (see Figure 6). For this, we simulate a personnel schedule under a predefined flight schedule, as well as given fault tolerance policies, and take note of requested numbers and qualifications of standby or reserve personnel. After an adequate number of simulation runs, we thus can recommend standby policies for each airport and time slot. A further aim is to assess different scenarios' impact on schedules to reveal consequences of temporary resource losses, e.g. damaged runways or raised probabilities of staff shortage in certain personnel clusters.

The simulation system currently under development is based on the event-based simulation approach (as described in [1]). Here, events of certain types yield state changes, which manifest at discrete points in time. The events are administrated in a priority queue, ordered by the time stamp of their occurrence. In a loop, the simulation engine extracts the event with the lowest time stamp, advances the simulation time accordingly, processes the event, updates the affected entities' states, and generates appropriate follow-up events, which are again entered into the priority queue. This is repeated until the priority queue is empty, i.e. all scheduled actions of the operational period are processed, and no more follow-up events are generated.

Using this technique, scenarios and occurrences of rare events can be handled by injecting corresponding simulation events into the priority queue prior to the simulation run.

In our simulation system airplanes encapsulate most of the simulation dynamics. Planes change their state at events like landing or opening doors. Main attributes are specified by the plane type, which holds functions for capacity, number and position of doors, avionic capabilities, etc. Combined with requirements of flight types, e.g. the number and qualifications of flight attendants, the demand for personnel is calculated. While processing these state changes, the simulation engine takes note of statistical data about delays and dynamic requests for standby personnel.

2 Conclusion and Future Work

In this paper we presented our project currently in development on robust airline job pairing. After explaining the context of the general airline operations planning process we gave an insight into the modules of the intended project architecture. The models and approaches presented differ from other models of airline personnel planning by considering single crew members instead of whole crews. We illustrated the common graph structure of flight networks and its potential influence on the JPP.

Since this project is still in its beginnings a lot of work has yet to be done. At the moment the real-world model, the setup of the database and a robust optimization program are refined in parallel. We look forward to our next milestone, the completion of the flight network and flight schedule generator, and to the comparison of the potentials of different graph structures on options of robust job pairing.

Acknowledgments

This work was partially supported by *Rhein Energie Stiftung Jugend Beruf Wissenschaft* under grant number W-10-2-002. We thank the other members of our work group Daniel Lückerath and Ewald Speckenmeyer.

References

[1] J. Banks, J. S. Carson II, B. L. Nelson, D. M. Nicol. *Discrete-Event System Simulation.* Pearson, 2010.

[2] R. Borndörfer, U. Schelten, T. Schlechte, S. Weider. *A Column Generation Approach to Airline Crew Scheduling*. Operations Research Proceedings 2005, pp. 343-348. Springer, 2006.

[3] D. Chakrabarti, Y. Zhan, C. Faloutsos. R-MAT: *A Recursive Model for Graph Mining*. Proceedings of the fourth SIAM International Conference on Data Mining, pp. 442-446. Society for Industrial and Applied Mathematics, 2004.

[4] Commission of the European Union: *Regulation (EC) No 859/2008*. Subsections N, O and Q. Official Journal of the European Union, 2008.

[5] V. Dück, L. Ionescu, N. Kliewer, L. Suhl. *Increasing stability of crew and aircraft schedules*.Transportation Research Part C: Emerging Technologies, Vol. 20, Issue 1, pp. 47-61. Elsevier Ltd., 2011.

[6] R. Elmasri, S. B. Navathe. *Fundamentals of Database Systems*. Pearson International Edition, pp. 57-129. Addison-Wesley, 2007.

[7] B. Gopalakrishnan, E. L. Johnson. *Airline Crew Scheduling: State-of-the-Art*. Annals of Operations Research, Vol. 140, pp. 305-337. Springer, 2006.

[8] G. W. Graves, R. D. McBride, I. Gershkoff, D. Anderson, D. Mahidhara. *Flight Crew Scheduling*. Management Science, Vol. 39, No. 6, pp. 736-745. The Institute of Management Sciences, 1993.

[9] S. Lavoie, M. Minoux, E. Odier. *A new approach for crew pairing problems by column generation with an application to air transportation*. European Journal of Operational Research, Vol. 35, Issue 1, pp. 45-58. Elsevier Science Publishers B. V., 1988.

Submitted: March 2011
Revised: June 2012
Accepted: August 1, 2012

Cognitive Aspects of Traffic Simulations in Virtual Environments

Sven Seele[1 *], Thomas Dettmar[1], Rainer Herpers[1,2,,3], Christian Bauckhage[4,5], Peter Becker[1]

[1] Bonn-Rhein-Sieg University of Applied Sciences, Dept. of Computer Science, Grantham-Allee 20, 53754 Sankt Augustin, Germany; *sven.seele@h-brs.de

[2] York University, Dept. Computer Science and Engineering; [3] Univ. of New Brunswick, Dept. Computer Science,

[4] Bonn-Aachen Int. Center for IT, Univ. of Bonn; [5] Fraunhofer Inst. for Intelligent Analysis and Information Systems IAIS

Abstract. Using virtual environment systems for road safety education requires a realistic simulation of road traffic. Current traffic simulations are either too restricted in their complexity of agent behavior or focus on aspects not important in virtual environments. More importantly, none of them are concerned with modeling misbehavior of traffic participants which is part of every-day traffic and should therefore not be neglected in this context. We present a concept for a traffic simulation that addresses the need for more realistic agent behavior with regard to road safety education. The two major components of this concept are a simulation of persistent agents which minimizes computational overhead and a model of cognitive processes of human drivers combined with psychological personality profiles to allow for individual behavior and misbehavior.

Introduction

Traffic simulations for virtual environments are concerned with the behavior of individual simulated traffic participants. The complexity of behavior in these simulations is often rather simple to abide by the constraints of processing resources. In traditional traffic simulations - used for road planning or traffic jam prediction - the behavior of individual traffic participants is also modeled, but the focus lies on the emerging overall behavior of the entire system, e.g. to identify possible bottle necks of traffic flow [29].

One objective of the FIVIS project [14,15] is to create a realistic bicycle simulator for road safety training of children. For that the traffic agents need to be persistent and react realistically to changing environmental conditions in real-time. Within virtual environments, traffic simulations need to be realistic only within a certain range of the visualization setup's viewing frustum. If simulated content is not directly visible, there is no need to spend major computing resources on its calculation.

Training simulators are widely used to increase safety in almost all areas of modern transportation (car, train, plane, etc.). With their help trainees are taught how to behave in dangerous situations without the risk of causing physical harm. The FIVIS project attempts to apply this advantage to bicycles, as there seem to be no such simulators commonly available today. Figure 1 shows the current system. So far dangerous traffic situations are realized in general by scripted events, defined for each car or any other traffic agent. This process is tedious, inflexible and might require trainees to follow a specific route. Additionally, subjects will memorize scripted events. Thus, they will not learn how to generally react to dangerous events, but rather how to react to specific actions performed by agents in certain situations at predetermined locations.

Figure 1. The FIVIS bicycle simulator is an immersive multi-screen system providing consistent 3D information to the visual field of subjects. Input (steering, acceleration, deceleration) is provided using a real bicycle fixed to the ground.

Thus, for the simulation of traffic agents in virtual environments, a cognitive traffic modeling approach is proposed that combines techniques from the field of traffic research and cognitive architecture research to address the stated challenges within a project called AVeSi

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 22(2), 2012, 83-88 | doi: 10.11128/sne.22.tn.10127

('Agentenbasierte Verkehrssimulation mit psychologischen Persönlichkeitsprofilen' – engl. 'Agent-based traffic simulation with psychological personality profiles').

This contribution will present a concept which aims at achieving the criteria stated above. It is structured as follows. The next section will summarize existing approaches. Following is the presentation of the developed concept that includes the discussion of how to achieve persistence in the simulation in Section 2 and the discussion of how to model cognitive traffic agents in Section 3. Finally, conclusions are drawn.

1 Related Work

The AVeSi project aims at developing an autonomous traffic simulation as an extension to the FIVIS bicycle simulator. The concept developed to fulfill this objective is based on previous research by Kutz et al. [17] who planned to utilize psychological personality profiles for traffic agents in virtual environments. The strict computational limits of virtual environment applications, such as digital games, often restrict the complexity of agent behavior in this domain. This fact has caused developers to utilize rather simplistic and static methods. Only in exceptional cases is increased realism attempted by utilizing adaptive artificial intelligence (see e.g. [1] and [27]).

In the field of cognitive architecture research, the ambitious goal is to create artificial intelligence able to solve general problems. Numerous attempts have been made to achieve this goal with varying concepts. The most prominent examples are the *Soar cognitive architecture* and ACT-R. The major difference between both is the type of memory and learning used. Soar is a *symbolic architecture*, which means an agent represents its environment and reasons about it using symbols [18]. ACT-R is an approach which uses elements of symbolic nature as well as elements of *emergent architectures*. Emergent architectures do not symbolize the knowledge of an agent, but instead utilize networks of distributed processing elements, typically emulating human brain structures [4,6]. Another example, called ICARUS [19], was already applied to the domain of in-city driving. The application was mainly concerned with basic driving behavior such as lane alignment or deceleration before turns. In [21] a cognitive architecture was introduced to simulate attention mechanisms of pilots and drivers to test assistance systems.

However, despite modeling erroneous behavior, its influence on decision-making was not addressed.

In contrast to traffic simulations in virtual environments, traditional traffic simulations have been researched since the 1930s. In [29] an overview of various approaches is provided. In this field, the behavior of a single traffic agent is usually not of interest to the researchers. Nevertheless, detailed and evolved descriptions of basic driving behavior, like the Intelligent Driver Model (IDM) [28], provide numerous starting points for cognitive traffic agents.

The field of traffic simulation also uses input from other disciplines. For example, in queueing theory, the formation and behavior of queues is analyzed mathematically. This includes the process of arriving at the back of the queue, waiting in the queue as well as being serviced at the beginning of the queue. Basic components of a queueing model are the queue and a servicing station. An example often used in the literature is a post office (e.g. [3]). Customers queue to be serviced at a counter. The time spent in the queue is dependent on the queue's length and the servicing rate of the servicing station. This queueing model can also be used to simulate traffic. Here the roads are modeled as queues and the servicing stations are junctions. Heidemann [12] and Vandaele [30] both developed basic frameworks for queueing-based traffic simulations. In [9] Gawron introduced a simple queueing model called *FastLane*, which is capable of modeling spill-backs by restricting the amount of vehicles in a queue and the amount of vehicles that can leave the queue.

2 Persistent Traffic Agents

An obvious way to save computational resources in traffic simulations for virtual environments is to remove traffic agents from the simulation once they leave the user's field of view (see Figure 2). This approach can lead to situations which are irritating to a user, but simulating all agents at all times would be too costly. Thus, the idea is to combine a detailed microscopic simulation within the user's vicinity with an additional coarser simulation level for non-visible traffic agents.

2.1 Queuing-based Traffic Simulation

At first, methods like treating traffic as fluid and applying smoothed particle hydrodynamics (SPH) to approximate common numerical solutions (cf. [23] and [24]) or applying macroscopic traffic models (e.g. the Lighthill-

Whitham-Richards model [20,22]) were investigated. Both approaches proved to be unsuited for the needs of the AVeSi project. Therefore, it is proposed to use a microscopic traffic model based on a queue-representation for the second simulation level (cf. [5,8,10,26,30,31]).

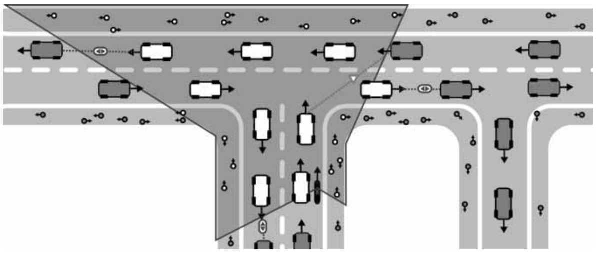


Figure 2. To save computational resources, traffic simulations in virtual environments usually simulate only agents within the user's field of view (white). The goal is to also simulate non-visible agents (gray) with less detail to achieve persistence with minimal computational overhead (image based on [17]).

As a queueing model, the *FastLane* model of Gawron has been chosen because it is a simple approach that provides meaningful approximations of travel times similar to those calculated by the well-known Nagel-Schreckenberg model as well as the inclusion of spill-backs [5,9]. Here the road network is represented by a directed graph $G = \{V, E\}$; where vertices V represent junctions and edges E represent roads. Each is modeled by a "physical queue", meaning a queue with limited space. Figure 3 presents an example for such a queue-based network. Each queue representing a road features a capacity C (maximum amount of vehicles that can leave a queue in one simulation time step), a length L, a number of lanes n_{lanes}, and a free flow velocity v_0 (the desired velocity a vehicle can drive or the maximum velocity allowed on that road).

When a vehicle enters a road, the vehicle's length of stay on this road is calculated. This time is called *travel* or *duration time* and depends on the free flow velocity and the current utilization of the road. After this time, the vehicle can reach a junction. Whether it does in fact reach the junction depends on the capacity of the road and the space on the junction.

In an initial approach, junctions will be modeled as either signalized or unsignalized and vehicles will be distributed randomly to connected roads. More sophisticated strategies to distribute vehicles at a junction can be found in [11] or [13].

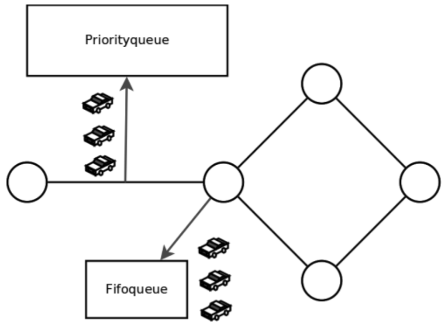


Figure 3. An illustration of how vehicles are represented in a queue-based road network. Roads are modeled as priority queues, where the priority is given by a vehicle's duration time. Junctions are servicing stations containing FIFO queues for each outgoing road [9].

When a vehicle enters a road, the vehicle's length of stay on this road is calculated. This time is called *travel* or *duration time* and depends on the free flow velocity and the current utilization of the road. After this time, the vehicle can reach a junction. Whether it does in fact reach the junction depends on the capacity of the road and the space on the junction. In an initial approach, junctions will be modeled as either signalized or unsignalized and vehicles will be distributed randomly to connected roads. More sophisticated strategies to distribute vehicles at a junction can be found in [11] or [13].

2.2 Linking Simulation Levels

The challenge of the queueing-based approach lies in determining the exact spatial position of a vehicle on the road and on modeling how one vehicle passes another vehicle within the queue. Accurately determining the spatial position of a vehicle is important for transitioning the vehicle from the coarser simulation level into the finer simulation level for visualization. Further, it is important to determine the fringe between these two levels. The vicinity, in which agents are simulated with the highest detail, should be as small as possible to save computational resources. On the contrary, it should be large enough to avoid unnecessary transitions between both levels and visual inconsistencies like sudden appearances of agents within the visual field.

Passing vehicles have to be incorporated into the simulation to be able to take different desired velocities and special vehicle behavior into account, e.g. a post office truck that stops every few hundred meters. Such vehicle behavior is necessary as disruptive factor for regular traffic, creating potential for risky decisions and dangerous situations.

3 Cognitive Traffic Agents

To achieve a realistic traffic simulation, artificial agents should not only be persistent within the simulated environment, but also need to show misbehavior like their human counterparts. However, they must do so only in situations where it really makes sense to an observer. Therefore, the authors believe that it is not enough to deviate from "normal" behavior by introducing fuzzy logic or random events, which is done frequently (e.g. see [7]). Instead an agent must consider its current perceivable situation when determining its actions. While doing so, it should also have the ability to calculate and take risks.

For this purpose the relevant cognitive processes should be modeled to induce realistic human behavior in traffic (e.g. perception, anticipation, decision making, etc.). Particularly modeling the influence of risk propensity on action selection will be a central aspect of the project. An agent choosing a risky action to reach its goals could cause it to break a traffic rule. Such misbehavior would result in potentially dangerous situations for other traffic participants such as a trainee using the FIVIS simulator.

3.1 Psychological Personality Profiles

The foundation of the agents' behavior will be a psychological personality profile as suggested in [17]. Such profiles will be based on the "Five Factor Model (FFM)" from psychology. According to this model, a human personality can be defined by a set of five distinct character traits: openness, conscientiousness, extraversion, agreeableness, and neuroticism. To determine a personality, a score for each trait can be extracted from standardized tests such as the Neo-Five-Factor Inventory (NEO-FFI) [2].

In [16] Herzberg argues that it is difficult to directly correlate driving behavior with the personality traits of the FFM. However, he found that such a correlation exists for the personality prototypes *undercontrolled*, *overcontrolled*, and *resilient*. Based on Herzberg's research a specific traffic situation has been designed for evaluation purposes that cannot be handled by regular, rule-based agents typical for virtual environments. In this scenario, a closed road system with an uncontrolled four-way intersection was modeled and traffic was simulated using an implementation of the Intelligent Driver Model (cf. Figure 4).

Agents arriving at the intersection need to yield to agents coming from the right. If, for example, four agents arrive at the intersection from each direction, each has to yield to another one resulting in a deadlock. The scenario described above, provided a road of about one kilometer in length with lanes in each direction. The maximum capacity of the network was about 500 vehicles. As illustrated by Figure 5, even at 3% of the maximum capacity (15 vehicles) it took only 333 seconds on average for a deadlock to occur. At 5% capacity and above, deadlocks appeared after less than one minute. The average and median times depicted in Figure were based on 15 simulations for each specified number of vehicles (15, 20, 25, 30, and 35) each starting with a random distribution of vehicles within the road network.

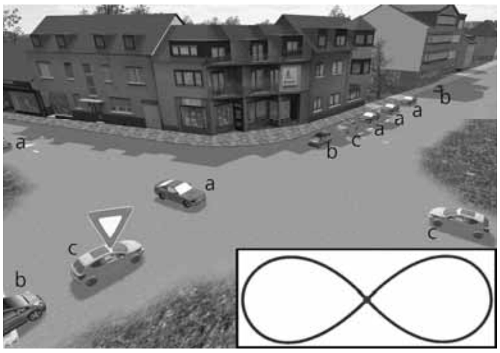


Figure 4. To evaluate agents with personality profiles, a closed road system with an uncontrolled four-way intersection was modeled. Vehicle types represent psychological prototypes: *undercontrolled* (a), *overcontrolled* (b), and *resilient* (c). The scene depicts a resilient driver waiving its right of way (indicated by a yield sign) allowing the driver on its left to cross the intersection, which resolves the deadlock (image based on [25]).

To resolve the issue we have augmented the agents with a mechanism to perceive such deadlocks (including a basic model of visual perception) and a personality profile assigning them to one of the three personality prototypes. According to Herzberg's findings we determine which of the four agents gives up its right of way based on its prototype. With these modifications, deadlocks can be detected and resolved, allowing the simulation to keep traffic flowing (cf. Figure 4).

Further detail and results which can be found in [25] indicate how personality profiles can be used to model individual behavior. In future revisions of the current realization, the profiles should also contain a dynamic component such that the behavior influenced by the agent's personality can be altered during the simulation, for example based on prolonged waiting times.

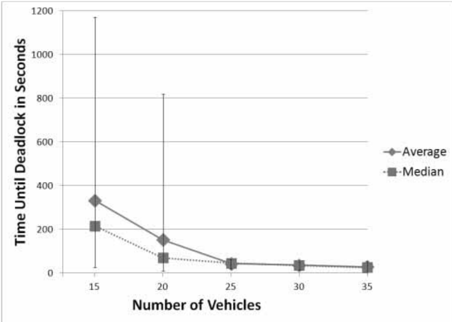


Figure 5. Average and median times until the simulation of the uncontrolled four-way intersection resulted in a deadlock. Vertical lines depict the longest and shortest times. (Image based on [25]).

3.2 Cognitive Processes

To further increase realism of agent behavior the application of ideas from cognitive architecture research is anticipated to yield the desired results. While the ultimate goal of this field is far more ambitious than what is necessary for the project presented here, the general ideas should provide an ideal starting point for cognitive traffic agents. Figure 6 illustrates an example of a cognitive traffic agent. It features typical modules from cognitive architecture research as well as an underlying personality profile based on the FFM (cf. Section 3.1). The most important aspects of such an agent are its links to the surrounding virtual environment, i.e. its perception and its actions.

In a first attempt to model human perception, a field of view was modeled using a geometric representation within a 3D game engine. Only agents and objects which enter this geometry are visible to the agent if they are not occluded by other objects. In the future this simple model could be extended by concepts of attention selection and attention division as presented in [21] and even audio cues. Especially by incorporating attention mechanisms, other modeled cognitive aspects, like emotion or the memory, could also influence the agent's perception. Additionally, as is the case in reality, the driving behavior might also be influenced by additional factors, such as age, gender, time of day, weather, road properties, and more. Most importantly, in combination with the personality profile, the modeled processes – in particular the decision module – should provide a means to encourage agents to take certain risks while trying to achieve their goals.

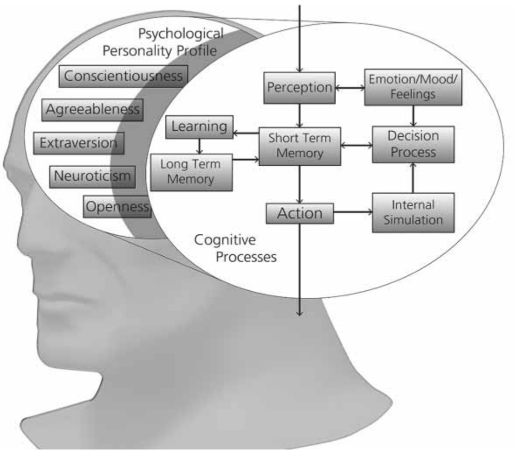


Figure 6. A possible model of a cognitive traffic agent for virtual environments containing the depicted modules from cognitive architecture research. An additional layer with a psychological personality profile may influence all cognitive processes (image based on [25]).

4 Conclusions

In conclusion, modeling persistent traffic agents whose behavior is inspired by human cognitive processes might improve road safety training applications, since they reflect real world traffic conditions more realistically. Especially, the ability to perform risky actions leading to agents breaking traffic rules and creating dangerous situations for other traffic participants should increase the realism of such a simulation. To achieve this goal, many issues remain to be solved. For example, what is the most suitable way for an agent to represent its surroundings internally or does an agent need to learn during the simulation? Other problems are of practical nature, like the question of how much overhead is created by constantly transferring agents from one simulation level to the other. Realizing and evaluating the current ideas presented in this article might be challenging, but if successful, other driving/traffic simulators and in particular traffic simulations in digital games will benefit from this research as well.

Acknowledgments

The FIVIS project received additional funding from the Deutsche Gesetzliche Unfallversicherung (grant FP-307). The AVeSi project is being funded by the FHprofUnt program of the Federal Ministry of Education and Research (grant 17028X11).

References

[1] Bakkes, S., Spronck, P., van den Herik, J. Rapid and reliable adaptation of video game AI. In *IEEE Transactions on Comp. Intelligence and AI in Games* 1, 2 (June 2009), 93 – 104.

[2] Borkenau, P., Ostendorf, F. *NEO-Fünf-Faktoren Inventar (NEO-FFI) nach Costa und McCrea*, Hogrefe, 1993.

[3] Bungartz, H.-J., Zimmer, S., Buchholz, M., Pflüger, D. *Modellbildung und Simulation*, Springer, 2009.

[4] Byrne, M. D.: *The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications.* Lawrence Erlbaum, 2003, ch. Cognitive Architecture, pp. 97-117.

[5] Cetin, N., Burri, A., Nagel, K. A large-scale agent-based traffic microsimulation based on queue model. In *Proc. of Swiss Transport Research Conf.* (Monte Verita, 2003).

[6] Duch, W., Oentaryo, R. J., Pasquier, M. Cognitive architectures: Where do we go from here? In *Proc. of the 2008 conf. on Artificial General Intelligence 2008* (Amsterdam, 2008), IOS Press, pp. 122-136.

[7] Fellendorf, M., Vortisch, P. *Fundamentals of Traffic Simulation.* Springer, 2010, ch. Microscopic Traffic Flow Simulator VISSIM, pp. 63 – 93.

[8] Gawron, C. An interactive algorithm to determine the dynamic user equilibrium in a traffic simulation model. In *International Journal of Modern Physics C* 9, 3 (1998)

[9] Gawron, C. Simulation based traffic assignment – Computing user equilibria in large street networks, PhD thesis (Cologne 1999).

[10] Grether, D., Neumann, A., Nagel, K. Simulation of urban traffic control: A queue model approach. Submitted to *The 1st International Workshop on Agent-Based Mobility, Traffic and Transportation Models, Methodologies and Applications* (2012).

[11] Heidemann, D. Queue length and delay distributions at traffic signals. In *Transportation Research – B* 28, 5 (October 1994), 377-389.

[12] Heidemann, D. A queueing theory approach to speed-flow-density relationships. In *Proc. of the 13th Int. Symposium on Transportation and* Traffic *Theory* (Lyon, France, 1996).

[13] Heidemann, D., Wegmann, H. Queueing at unsignalized intersections. In *Transportation Research – B* 31, 3 (June 1997), 239 – 263.

[14] Herpers, R., et al. *Multiple Sensorial Media Advances and Applications: New Developments in MulSeMedia*, IGI Global, 2011, ch. Multimedia Sensory Cue Processing in the FIVIS Simulation Environment, pp. 217-233.

[15] Herpers, R., et al. FIVIS – A Bicycle Simulation System, In *World Congress on Medical Physics and Biomedical Engineering* (WC 2009), IFMBE Proc. Vol. 25/4 (Berlin, 2009), Springer, pp. 2132 – 2135.

[16] Herzberg, P. Beyond “accident-proneness”: Using Five-Factor Model prototypes to predict driving behavior, In *J. of Research in Personality* 6, 43 (2009), 1096 – 1100.

[17] Kutz, M., Herpers, R. Urban traffic simulation for games, In *Future Play '08 Proc. of the 2008 Conf. on Future Play: Research, Play, Share* (NY, 2008), ACM, pp. 181-184.

[18] Laird, J. E. Extending the Soar cognitive architecture. In *Proc. of the 2008 conference on Artificial General Intelligence 2008* (Amsterdam, 2008), IOS Press, pp. 224 – 235.

[19] Langley, P., Choi, D. A unified cognitive architecture for physical agents. In: *Proc. of the National Conference on Artif. Intelligence* (2006), AAAI Press, pp. 1469 – 1474.

[20] Lighthill, M. J., Whitham, G. B. On kinematic waves: II. A theory of traffic on long crowded roads. In *Proceedings of the Royal Society A* (London, 1955), Vol. 229, No. 1178, pp. 317 – 345.

[21] Lüdtke, A., Weber, L. Osterloh, J., Wortelen, B. *Digital Human Modeling.* LNCS, vol. 5620, Springer, 2009, ch. Modeling pilot and driver behavior for human error simulation, pp. 403-412.

[22] Richards, P. Shock waves on the highway. In *Operations Research* 4, 1 (Feb. 1956), 42-51.

[23] Rosswog, S., Wagner, P. “Car-SPH”: A lagrangian particle scheme for the solution of the macroscopic traffic flow equations, In *Traffic and Granular Flow ’99* (Stuttgart, 1999), pp. 401 – 407.

[24] Rosswog, S., Wagner, P. Towards a macroscopic modeling of the complexity in traffic flow, In *Phys. Rev. E Stat. Nonlin. Soft Matter Phys* 65, 3 Pt 2A (February 2002).

[25] Seele, S., Herpers, R., Bauckhage, C. Cognitive agents for microscopic traffic simulations for virtual environments. To appear in *Entertainment Computing – ICEC 2012*.

[26] Simon, P., Esser, J., Nagel, K. Simple queueing model applied to the city of Portland. In *International Journal of Modern Physics C* 10, 5 (1999), 941 – 960.

[27] Thurau, C., Bauckhage, C., Sagerer, G. Learning human-like movement behavior for computer games. In *Proc. Int. Conf. on the Simulation of Adaptive Behavior* (2004), MIT Press, pp. 315 – 323.

[28] Treiber, M., Hennecke, A., Helbing, D. Congested traffic states in empirical observations and microscopic simulations. In *Phys. Rev. E* 62, 2 (August 2000), 1805 – 1824.

[29] Treiber, M., Kesting, A. *Verkehrsdynamik und – simulation*, Springer, 2009.

[30] Vandaele, N., van Woensel, T., Verbruggen, A. A queueing based traffic flow model. In *Transportation Research D* 5, 2 (March 2000), 121 – 135.

[31] Van Woensel, T., Vandaele, N. Modelling traffic flow with queueing models: A review. In *Asia-Pacific Journal of Operations Research* 24, 4 (August 2007), 435 – 461. http://dx.doi.org/10.1142/S0217595907001383

Submitted: March 2012
Revised: June 2012
Accepted: August 1, 2012

Network - wide Evaluation of Cooperative Traffic Systems using Microscopic Traffic Flow Simulation

Jonas Lüßmann*, Claudia Sánta

TUM Technische Universität München, Chair of Traffic Engineering and Control, Arcisstrasse 21, 80333 Munich;
*jonas.luessmann@vt.bv.tum.de

Abstract. The eCoMove project, which is funded by the European Commission under the 7th Framework Programme Research and Technological Development, creates a system to reduce the CO_2 emissions and fuel consumption of vehicles. The system uses the cooperative technology of V2I-Communication (vehicle to infrastructure communication). Within the different subprojects, in-vehicle and infrastructure site applications are developed. As only a limited number of vehicles are equipped with the system, network wide effects of the systems cannot be measured within the real test sides. A validation of these effects can be done using microscopic traffic flow simulation. This article introduces the concept of the simulation environment used for the validation within the ecoMove project.

Introduction

The aim of the eCoMove project is to achieve a reduction of 20% of CO_2 emissions and fuel consumption for private and transport vehicles. By providing communication between vehicles and infrastructure, drivers can receive information as to how to drive more efficiently on urban and interurban streets.

On one hand, drivers receive information on an optimal speed when they approach a traffic signal. On the other hand, infrastructure site applications use extended floating car data (eFCD) to improve the signal control.

For single vehicles it is quite easy to validate the effects. However, due to the small number of vehicles equipped with the eCoMove system within the project, it is not possible to validate a network wide effect of the system in reality. Therefore, the microscopic traffic flow simulation VISSIM [1] is used to support the development and evaluate the entire system. The simulated network is based on Munich and Helmond (Netherlands) and is calibrated with real traffic data.

To provide the infrastructure site ecoMove system with realistic traffic data, several adapters are developed to convert simulation data into messages that can be received by the eCoMove system, as in reality. This was done to make it possible to substitute the real test site by the simulation environment.

The article describes the general concept of the extended simulation environment used within the eCoMove project for development and evaluation. This is in detail the technical interfaces between the microscopic simulation and the infrastructure site eCoMove System, the modeling of the test site and the modeling of the in-vehicle systems and their calibration.

1 The eCoMove Project

The eCoMove project is funded by the European Commission under the 7th Framework Programme Research and Technological Development. Thirty two partners from ten countries are involved in the project. The partners mainly work as vehicle manufacturers, automotive suppliers, communication providers, map providers, infrastructure suppliers or at research institutions. The project started in April 2010 and will end in September 2013, with a final event in Aachen.

The project is divided up into six subprojects. In addition to project management, the sub-projects are:

- *"Core Technology Integration"*, for the development of core technologies such as the communication platform, the communication protocols, the data bases, the digital map and several traffic models,
- *"ecoSmart Driving"*, for the development of in-car applications for the support of navigation and longitudinal driving behaviour,
- *"ecoFreight & Logistics"*, for the development in-truck systems for the support of navigation and longitudinal driving behavior and central site tour planning,

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 22(2), 2012, 89-94 | doi: 10.11128/sne.22.tn.10129

- "*ecoTraffic Management and Control*", for the development of infrastructure site applications for central site routing and traffic light control and
- "*Validation and Evaluation*", for impact assessment and evaluation of the whole system.

Due to the low number of equipped vehicles, it is only possible to assess the impact of single vehicles in reality. Therefore microscopic traffic flow simulation is used for a network wide impact assessment.

2 Concept of Simulation Environment

On the one hand, the microscopic traffic flow simulation should provide data for testing and parameterization of the applications developed in the subproject „ecoTraffic Management and Control". On the other hand, it should also be used for a network wide impact assessment of the whole system.

To fulfill both requirements, a software-in-the-loop approach was chosen (see Figure 1). The software implementation for the infrastructure system is exactly the same for simulation and real test site. The simulation represents the reality and can therefore replace it.

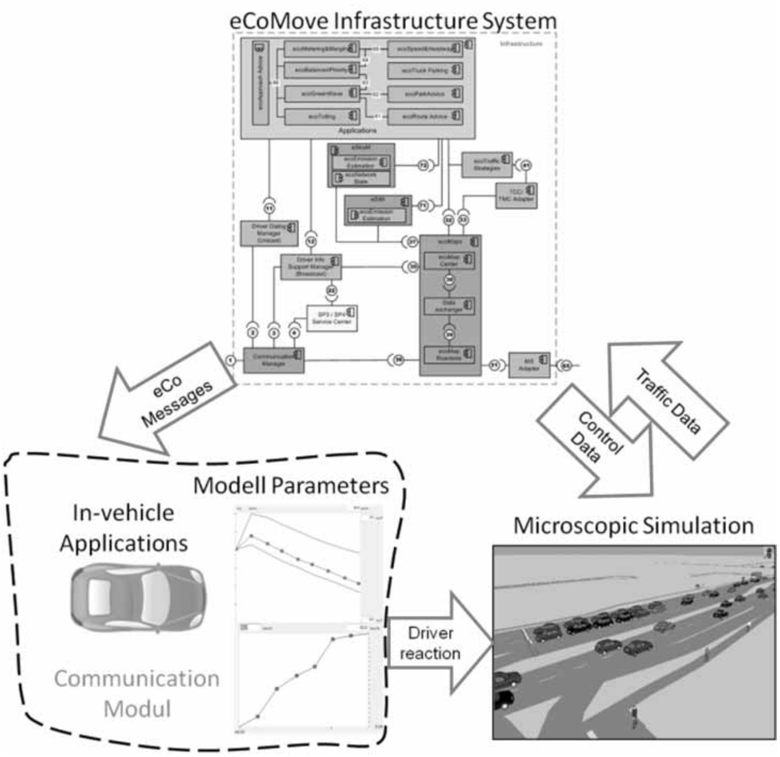


Figure 1. Concept of the eCoMove simulation environment [2].

Via adapters, the simulation provides traffic data for the eCoMove infrastructure system. The data from the virtual vehicles and infrastructure sensors is also converted into the same format as in reality. In return the traffic light control data and the driver reactions due to the recommendations given to the drivers are represented in the traffic flow simulation.

In contrast, the in-vehicle applications and the communication are not simulated but represented by models. A main reason for that is that due to computational power limits, it is not possible to connect all single eCoMove in-vehicle applications to the simulation environment. The microscopic traffic flow simulation tool VISSIM [1] is used. VISSIM uses a stochastic, time step based model. The driver-vehicle-unit is the basic unit of this model.

3 Linking Infrastructure System - Simulation Environment

In the following section, the link between the simulation and the infrastructure system of eCoMove is described. For that purpose, an overview of the architecture and the VISSIM specific communication modelling is described.

3.1 Architecture Overview

An overview of the architecture is given in Figure 2. On the left side, the simulation environment is shown and on the right side, the infrastructure system, which can be run on a road side unit or on a central server is depicted. While the left side will be replaced by the real test site environment, the right side remains. Two example applications are selected for illustration purposes: ecoGreenWave as an example for the group of applications dealing with signal control, and ecoApproachAdvice as an example for the group of applications used for communication with the vehicle.

The system is composed of many components with loose connections. The microscopic traffic simulation, VISSIM (representative for the real test sites) is connected with the ecoMap through many interfaces. The ecoMap represents the central data repository. All static map data and dynamic data generated over time can be accessed via the ecoMap.

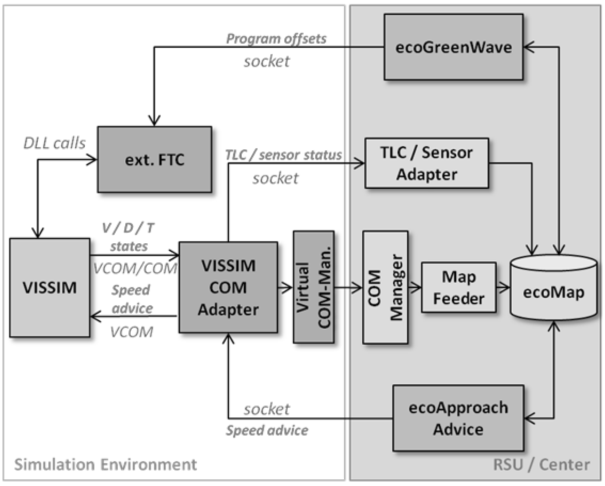


Figure 2. Architecture of the linkage simulation-infrastructure system adapted from [3].

It is not implemented as a data base for data storage. It provides exchange of transient data for all applications.

The applications are implemented as OSGi (Open Services Gateway initiative) services. They communicate through a message service called ecoMessages service. If an application is interested in data from another application, it has to sign up for the corresponding for message. It gets the message as soon as it is available.

3.2 Modelling the Communication using VCOM

The communication between vehicles and vehicles or vehicles and infrastructure is modelled by the VCOM module. It is implemented as dynamic link library (dll). Given this implementation vehicle information can be requested efficiently using direct method calls. The information contains position, velocity, acceleration, safety distance, vehicle type and motion vector of the vehicles.

VCOM has a dedicated communication modelling component. The information is transmitted within a certain reception probability depending on the distance between sender and receiver and the number of communicating vehicles (see Figure 3). The reception probability is calibrated using the "Network Simulator ns-2". Communication can be modelled using the IEEE Standard 802.11p with different reception probability distributions or using UMTS/3G. In order to access and handle vehicle information VISSIM hands over control to the specific application after each simulation step. The application receives all information on communicating vehicles via the VCOM interface.

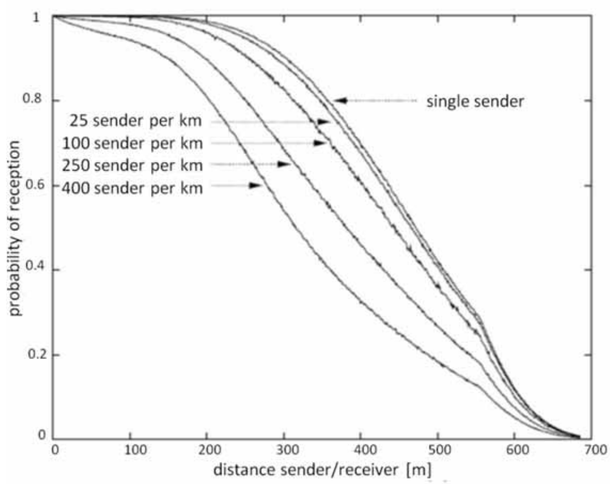


Figure 3. Reception probability depending on the distance between sender and receiver and the number of communicating vehicles per kilometer [4].

An internal logic of the application proofs if the current information could be exchanged by sending a request to the VCOM module and it proofs if any action should be triggered. If so, the action (e.g. velocity adaption) is executed and control is handed back to VISSIM.

3.3 Implementation

In order to connect the simulation environment to the infrastructure, primarily two interfaces are needed: One to access vehicle data such as position from simulation, and one to get information on signal control and detector state. Access to vehicle data is realized using the VCOM module in the VISSIM COM Adapter. The Virtual COM Manager simply trans-forms vehicle data into the message format defined in eCoMove. Access to signal control is done in the component Traffic Light Control by calling the appropriate dll of VISSIM.

The VISSIM COM Adapter and the Traffic Light Control (TLC) interface together form the necessary framework around the simulation environment to connect the simulation to the infrastructure system. The connection itself is realized by socket connections. Therefore, communication between both systems is facilitated, even though they are based on different programming languages (the simulation environment is written in C++ and the eCoMove infrastructure system is written in Java). It is even possible to run the simulation environment on a different server than the infrastructure side.

Vehicle data and signal control data flow both to the real test sites into the ecoMap where they are accessed by the applications. The applications process the relevant data and provide the results. The resulting data flows back to the simulation environment in form of signal control adoptions (ecoGreenWave) or velocity adaption (ecoApproach Advice). The applications can be tested using the feedback channel. It has to be noted that the simulation must run in real time, meaning one simulation time step corresponds to one real second, in order to ensure that the functionality of the overall system corresponds to the runtime in the test sites.

In addition, impacts of the applications can be evaluated. Driver behaviour and emissions change due to speed advisories at intersections or due to signal control adaption can be evaluated.

4 Test Site Modelling Munich

The test site covers the main parts of northern Munich, including the Highway A99 (see Figure 4). The size was chosen to make it possible to simulate all infrastructure site applications. Therefore the routing applications were responsible for the network size.

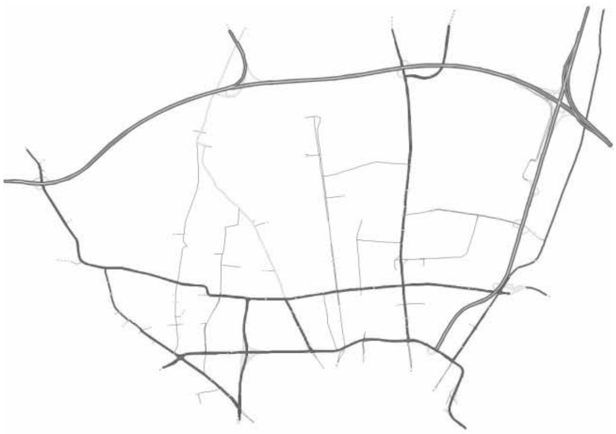


Figure 4. Network model from the macroscopic tool VISUM [5].

The network is generated from a geo database developed within the German project simTD [6]. From this database it is possible to export different networks for different simulation settings needed. Main objects of the database are the street network, sensors, traffic light control and the traffic demand including traffic volumes and routes. The traffic demand is calibrated based on the tool VISUM [5].

Both the geo database and the ecoMap are based on NAVTEQ data. Therefore the mapping is relatively easy. As the NAVTEQ data does not contain detailed enough data about the intersections, including traffic light control all intersections are modelled by hand (see Figure 5).

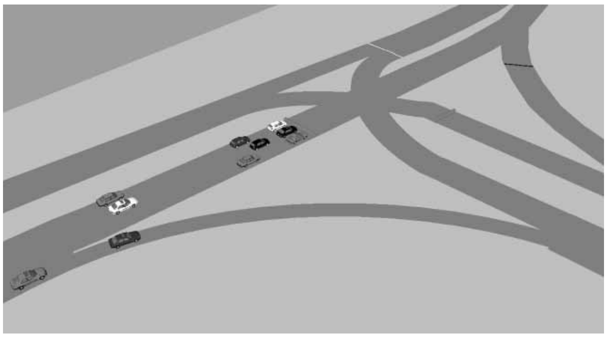


Figure 5. 3D model of a hand modelled intersection.

5 Modelling the Driver

The eCoMove system mainly influences two components of the driver behaviour: longitudinal vehicle movement and route choice.

5.1 Model for Longitudinal Vehicle Movement

For longitudinal vehicle movement, VISSIM uses a psycho-physical car following model [7][8]. The basic idea of the model is that the driver is always driving in one of the following four modes:

- *Free driving:* No influence of preceding vehicles is observable. The driver tries to reach his desired speed and to keep it.
- *Approaching:* The driver adapts his speed to the lower speed of a preceding vehicle. The driver decelerates so that the speed difference of to the preceding vehicles becomes zero when he reaches his desired safety distance.
- *Following:* The driver follows the preceding vehicle without accelerating or decelerating, trying to keep the safety distance constant. Due to imperfect throttle control and imperfect estimation the speed difference oscillates around zero.
- *Braking:* The driver decelerates if the distance falls below the desired safety distance. of the observed driver.

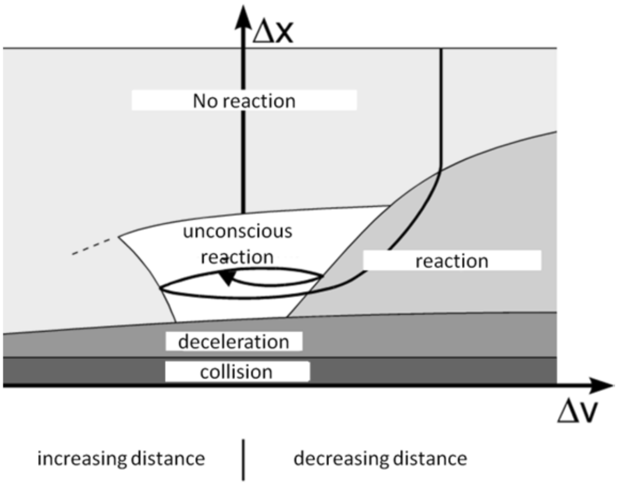


Figure 6. Car Following Model according to Wiedemann, 1974 [1].

Figure 6 shows the different modes as a function of relative speed and the distance to the preceding vehicle. In the area of no reaction the driver is in the free driving mode. As the distance is decreasing the driver changes to the approaching mode and finally to the following mode.

In the following, parameter values are listed for uninfluenced drivers. They are known from a number of measurements. For drivers influenced by the eCoMove system, no variations in these values were expected, because they only refer to surrounding traffic in direct vicinity and the eCoMove information given to the driver looks further ahead than the driver is able:

- *Look ahead distance* is the distance that a driver can see forward in order to react to other vehicles in front or beside.
- *Number of observed vehicles* affects how well vehicles can predict other driver behaviour and how to react accordingly.
- *The look back distance* defines the distance that a vehicle can see backwards in order to react to other vehicles behind.
- *Temporary lack of attention:* During this time vehicles will not react to a preceding vehicle.
- *Wiedemann model parameters:* Dependent on the chosen car following model.

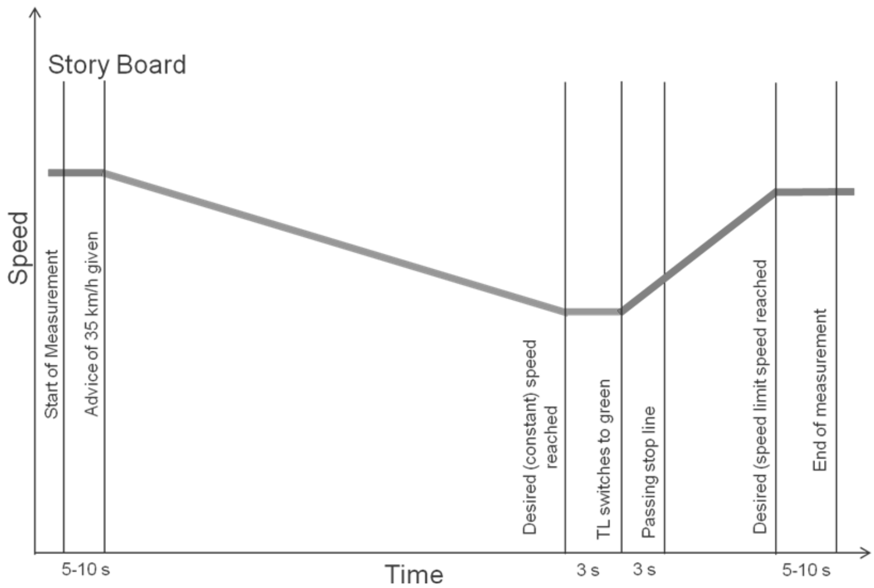


Figure 7. Story board of a traffic light approach for the calibration of VISSIM by a driving simulator study.

Further important input parameter for the car following model are distributions of:

- *Desired velocity,*
- *Desired acceleration and*
- *Desired deceleration.*

For uninfluenced drivers, measurement values are already available. For drivers influenced by the system, parameters are determined using driver simulator studies for different use cases.

For this issue special test runs are designed to directly measure the input parameter distributions for VISSIM - desired speed, desired acceleration and desired deceleration. Figure 7 shows an approach to a traffic light with a speed advice given for the driver.

A main requirement is that the driver is able to drive his desire speed, acceleration and deceleration. This means that he is only influenced by the onboard display. He should not be influenced by other things, like other vehicles as this would falsify the measurements.

As already mentioned, the influence on the longitudinal driver behaviour by other vehicles is already recognized by the car following model from Wiedemann.

5.2 Route Choice

The compliance rate of drivers getting a route advice depends on many factors, such as the navigation device, the number of alternative routes, and the length and travel time of the route choices.

Therefore, a calibration of route decisions through driver simulator studies does not make sense. For drivers who are not familiar with the place, a compliance rate of hundred percent can be assumed. To determine the compliance rate of local drivers, data of the research project wiki [9] are used. In wiki, the route choice behaviour of local drivers has been investigated in the north of Munich.

6 Conclusion

The microscopic simulation environment described above is used to parameterize, test and evaluate the infrastructural applications and their combination with the in-vehicle applications on a network level. It outputs traffic efficiencies and inefficiencies within the network as a direct result.

In addition assessment of different penetration rates is possible and we can study the effects of the eCoMove system on vehicles equipped with the system separately from the effects on non-equipped vehicles.

References

[1] A.P. Sage. *Dynamic Systems*. In D.P. Atherton, P. Borne, eds., Concise Encyclopedia of Modelling and Simulation, Oxford, 1992, 91 - 92.

[2] PTV AG. *VISSIM 5.40 User Manual*. Karlsruhe, 2011.

[3] J. Lüßmann, T. Schendzielorz, J. Vreeswijk. *Extension of Simulation Functionalities and Test Site Modelling*. eCoMove deliverable 5.3, 2011

[4] MAT.TRAFFIC. *Internal report from eCoMove*. 2011.

[5] M. Killat, F. Schmidt-Eisenlohr, H. Hartenstein, C. Rössel, P. Vortisch, S. Assenmacher, F. Busch. *Enabling efficient and accurate large-scale simulations of VANETS for vehicular traffic management*. Proceedings of the fourth ACM International Workshop on Vehicular Ad Hoc Networks (VANET), pp. 29-38, Montreal, Quebec, Canada, September 2007.

[6] PTV AG. *VISUM 12 User Manual*. Karlsruhe, 2012.

[7] MAT.TRAFFIC. *Dokumentation der Geo-Datenbank für Feldversuch und Verkehrssimulation in simTD*. Aachen, 2011.

[8] R. Wiedemann. *Simulation des Straßenverkehrsflusses*. Schriftenreihe des Instituts für Verkehrswesen der Universität Karlsruhe, issue 8, 1974.

[9] R. Wiedemann. *Modeling of RTI-Elements on multilane roads.* Advanced Telematics in Road Transport, DG XIII, Brussels, 1991.

[10] wiki – *Wirkungen von individueller und kollektiver ontrip Verkehrsbeeinflussung auf den Verkehr in Ballungsräumen*. Final report, 2012.

Submitted: March 2012
Revised: June, 2012
Accepted: August 1, 2012

Modelling and Simulation of Student Pedestrian Traffic at University Campus

Martin Bruckner[1,3 *], Shabnam Tauböck[1], Nikolas Popper[1,3], Stefan Emrich[2], Benjamin Rozsenich[1], Salah Alkilani[1]

[1] Inst. f. Analysis and Scientific Computing, [2] RED Real Estate Development, Vienna University of Technology
[3] dwh Simulation Services, Neustiftgasse 57-59, 1070 Vienna, Austria; **martin.bruckner@dwh.at*

Abstract. <more-space> is a software project of Vienna University of technology for supporting the planning phase of 'University2015' - a project to renovate all university buildings and to improve the existing infrastructure and the inherent processes. <more-space> determines and evaluates the spatial resources and time needs required and introduces a model for the room management that can simulate the usage of resources to optimize the planning of rooms and schedules. This contributions presents a part of the <more-space> project, the dynamic simulation for the student pedestrian traffic, which takes into account time requirements for reaching different lecture hall buildings and lecture halls on an individual basis, allowing also to incorporate needs for physical handicapped persons. Modelling and simulation of this dynamic individual student traffic is based on advanced cellular automata model, which are integrated into other <more-space> components (distributed DEVS model).

Introduction

The aim of the described simulation project as a part of the simulation system <more space> is to simulate the movement of students between lecture rooms, attending their regular curriculum to implement dynamic vacation times. Major outcome is the calculated time which they need to move from a starting point (for example an auditorium) to another location (arrival point).

The program is realized in the object-oriented programming language JAVA and connected to *Enterprise Dynamics*. Modeling approaches are Cellular Automata (CA), Agent Based Modelling (AB) and discrete simulation because the literature of this approach is widely spread, and after analysis of the project, the cost-benefit calculation for this modeling was the best. CA and AB approach can manage the dynamic behavior of the students finer and more efficient.

The dynamic model implemented in *Enterprise Dynamics* is the main model and simulation system including the data model, process descriptions and dynamic behaviour as using of resources depending on different system or environmental dependencies.

The described model for simulation of dynamic behaviour is implemented in the object-oriented programming language Java and connected via TCP/IP with Enterprise Dynamics. To model the dynamic behavior of single individuals an agent-based system was chosen in which the individuals move on a discrete grid. The cell size is 0.125m x 0.125m, so that one m^2 consists of 64 cells. Each student takes 4 x 4 cells, or 0.5m x 0.5m. The forward movement of people in a building depends on several interrelated factors. Some of these are e.g. the density of people in a group, or the maximum speed which varies for each individual. Of course these factors are conditioned by the environment; for example moving into a room area or a staircase makes a noticeable difference. It is also of crucial importance, whether a person is facing multiple other individuals. All this is relevant for the speed and thus for the required time which the students need to switch from location A to another location B. One special aspect of this project was to integrate the possibility to simulate the path for physical handicapped persons.

1 Modelling

Basically, this simulation consists of two simulators. On the one hand Enterprise Dynamics, a commercial Software from INCONTROL based on the DEVS (Discrete Event System Specification) formalism in which the optimization of the room utilization is modeled. On the other hand a proprietary development in JAVA which provides the times who are needed from the students to changed the lecture rooms.

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 22(2), 2012, 95-100 | doi: 10.11128/sne.22.tn.10131

This part shall ensure that the time between two teaching units, who are maybe in different buildings, is sufficiently dimensioned. In order to implement the task a cellular automats (CAs) model was used. The main components of this model is a discrete plane and the individual agents, or as in our case, the individual students who are moving on this grid, and whose decision on their further action depends on the surrounding agents. Because of the fact that the university area is too big to display to only one of such grids, the buildings are divided in logically coherent parts that are connected at several points to give the agent the possibilities to change these discrete planes. In order to allow the students to move through the huge number of planes in the shortest possible way a combination of graph theory and a kind of scalar field is used.

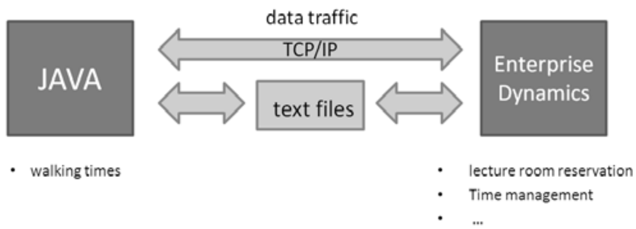


Figure 1. Data traffic.

2 Building Plans

The import of the building structures works via special edited planes. The original plans exist in the common used Autocad format .dwg. The editing is explained in the following shortly.

- Unnecessary information like dimensioning information, energy and water installations and many more are removed because it has no importance for the simulation and disturbs the clearness of the plans.
- The particular parts of the building will be colorized in one of the ten defined colors. (corridor, up- downstairs, wall, outdoor private or class room, no pass, window, text) This is necessary because the importer of the plans distinguishes the parts of the building only by their colors.
- In the next step the complete colored image will be exported to an image file in the Portable Network Graphics (PNG) format. The resolution of the graphics is chosen such that each pixel represents just 0,125m x 0,125m in reality, what exactly corresponds to the simulation of a cell of the CAs. For each pixel, the three primary colors of the RGB color space (red-green-blue) and in addition the alpha value are stored. The range of values for each of these four channels is 0-255, which is 8 bit consuming. This relative roughly resolution is also a reason why it is imported to remove the unnecessary and smallest details out of the plan as mentioned in the first point because this information degenerates after the export most to one dot.
- The last working step is to insert, with a special program, the so called Change Areas, Elevator Areas and Doors. This software tool is furthermore important because here the declaration of the buildings, CAs, rooms and many more will be performed and finally stored in the project file. This Comma-Separated Values (CSV) project file is in a way the "brain" of the data set.

The transformation of an original to an edit plan is depicted in the following pictures.

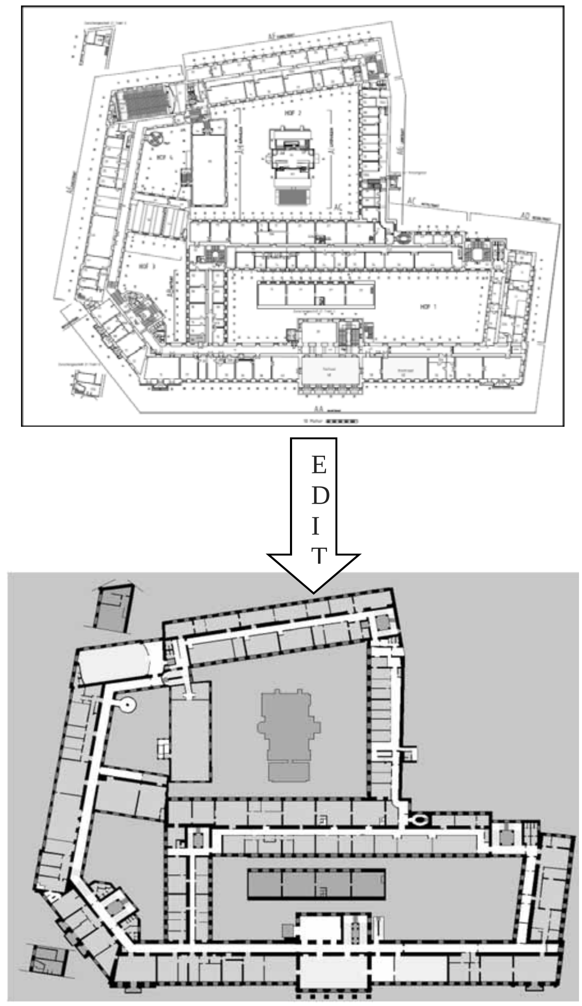


Figure 2. Example of a .png file (TU-Vienna, main building, first floor).

3 Routing

3.1 Global Routing

To simplify student´s life they need a plan how to get from point A to point B and if possible in the shortest way. To ensure that this happens, the graph theory was used. With aid of images and CSV files a connected undirected and weighted graph is generated.

- Node: Basically it exist only two types of nodes, doors and change-areas which includes also the elevator-areas. In case of doors we consider only the entrance of the lecture-rooms and building-exits. Doors, such as fire or corridor doors, are not in use. Only a small narrowing in corridor handicappes the persons. The change-areas give the individuals the ability to change the CAs, for example at the end of a stair or in case of an elevator to change the floors.
- Edges: In every CA each node is connected via edges with each other, whereby each edge has two weights, the distance in meter and the time which an average human needs for this way without obstructions.

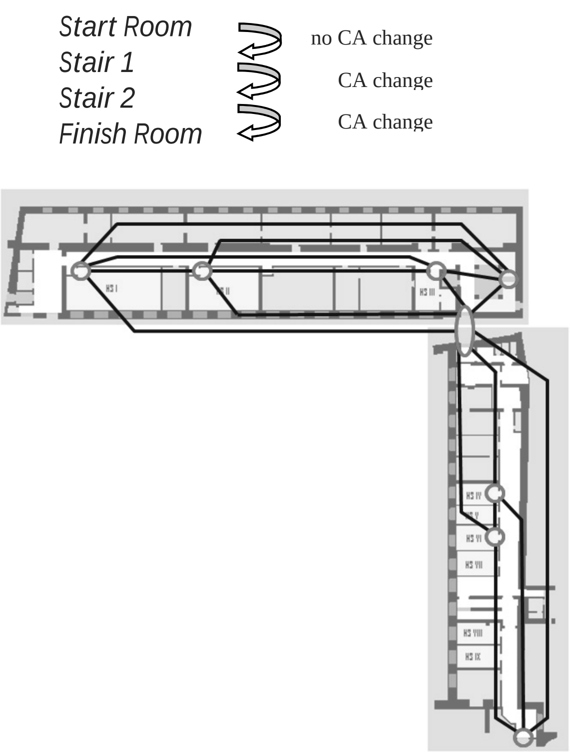


Figure 3. Example of a graph.

With the aid of this nodes and edges and a slightly modified Dijkstra's algorithm it is now possible to find a path, which has a good combination of shortest and fastest way, through the building.

Considering the shortest path is not always the best. For example when people have to wait for some time in front of an elevator instead of using the stairs. This algorithm provides now a list which looks like the following.

3.2 Local Routing

The aim of the local routing is to move the individuals in the simulation the shortest way through the current Cellular automata. To implement this, the simulator uses a Static Floor Field. As the name suggests, this is a field of static values, which contains the distances, to the various nodes, in cells. This field is embedded into the CAs. So it is possible to “ask” each cell: “How far is it to the door with name doorLectureRoom8?”

To determine now the shortest path, for each direction the associated cells will be scanned and afterwards they calculate the arithmetical average. So the direction with the smallest distance works in favor of the next step. To initialize this data structure, a modified variation of the Flood Fill algorithm is used. This is a simple procedure of Computer graphics; in which the bording areas are colored.

6	6	6	6	6	6	6	6
5	5	5	5	5	5	5	5
4	4	4	4	4	4	4	4
3	3	3	3	3	3	3	3
2	2	2	2	2	2	2	2
2	1	1	1	1	1	1	2
		0	0	0	0		
		-1	-1	-1	-1		

Figure 4. Static floor field.

Figure 4 shows an example from a static floor field and shall give a better view of the keynote. The yellow cells in the centre below illustrate a door bordered left and right from a part of a wall. A cell with the distance 0 indicates a starting point of the door. The remaining numbers show the distance of the associated cells to this starting points.

4 Deadlock

- React to other students: The first and perhaps most effective method is the timely reaction of contra flow or other barriers. To implement this, a searching area of approximately 1.5 meters wide and 8 meters in length will be scanned, while all found students are divided in three groups according to their relative walking direction; (1) traffic, (2) cross traffic, (3) oncoming traffic. This subdivision will be considered by the selection of the next step. For example, when overtaking a student, or make room for oncoming traffic.

- Jostle other persons: In order to allow a necessary change of direction even if another student who walks parallel and in the same direction and prevents jostling, each individual has the possibility to make a request to the neighbor to make room. The student can afterwards decide if he obeys this suggestion or not depending on his or her possibility.
- Right-hand walking: The next algorithm to avoid collision is the preferred passing of a contra flow on the right side. This behavior can be observed repeatedly in public and is also described in the literature by various experiments.
- Shrink Student size: Another very useful method is to imitate the behavior of people coming into the situation that the required space is too small for "normal" walk. We instinctively shrink our space, for example by rotating our shoulders. This shrink is also implemented in this model. Each student normally requires 0.25m2 or 4 x 4 cells in simulation. This space may, if the situation requires it, be reduced to 3 x 3 or 2 x 2 cells.
- Waiting in front of the auditorium: Further potential sources of blockages are the entrances of the lecture rooms. So arriving Students are waiting outside of the room until the room and therefore the doors are free.
- Second level: The last algorithm to resolve deadlocks is a second emergency level. About this second level blockade can be reduced and it will ensure that the students reach their specified destination.

5 Walking Speed

The walking speed of people is influenced by different factors. To simplify, the simulation acceleration and deceleration will be ignored.

5.1 Depending on density

A crucial factor of moving forward is the density of the traffic. The relation of density and speed is described with the so-called Kladek formula, which will be described in the following part.

v_h^0 Max. speed, depending on the underground.

ρ^{max} Max. density, 5.4 P/m^2

γ calibrate constant (empirical determined), 1.913.

$$v_h(\rho) = v_h^0(1 - exp^{\left(-\gamma\left(\frac{1}{\rho} - \frac{1}{\rho^{max}}\right)\right)})$$

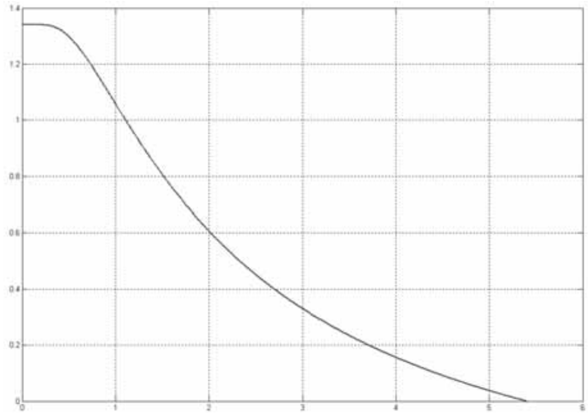


Figure 5. Relation of Kladek v_h^0 =1.34 m/s.

5.2 Depending on underground

The ground where the persons are moving is also an important factor. To simplify this model only three different types are defined. These values are the v_h^0 for the mentioned Kladek formula:

- Corridor 1.34 m/s
- Upstairs 0.61 m/s (only horizontal)
- Downstairs 0.71 m/s (only horizontal)

Because the size of the students varies, but it is in any case bigger than one cell, it is possible that they occupy cells with different speed-values, for example when entering a stair. To deal with this problem, the maximum speed v_h^0 is the arithmetic average of all cells.

5.3 Depending on random speed

In addition to this speed calculation each student has a random speed factor which is generated with a special probability distribution. This factor will be multiplied with the final speed value to bring the simulation closer to reality.

6 Calculation of the Next Step

Afterwards there exists a brief overview of the calculation of one time step. In the simulation one time step is 250ms. Please note that a time step is not the same as a cell shift. One time step has usually a few cell shifts.

- Data ascertainment: density, traffic, …
- Calculate next direction: with the aid of the collected data the next direction for the cell shift is chosen.

- Calculate speed and time: calculate the possible speed in this direction and the resultant time for this shift.
- Repeat 1-3: The first three points are repeated until the difference of the sum of the single shift time and the simulation time, in our case 250 ms, is minimal.

$$\left|\sum (shift\ times) - 250ms\right| = minimal$$

- Calculate discretization error: the difference of the above formula is the start value in the next iteration. This helps to minimize the discretization error.

The following example shows what happens without consideration of this discretization error.

Student speed = $1{,}34\ \frac{m}{s}$

Step time Δt = $0{,}25$ s

From this it follows a distance of

$$1{,}34\frac{m}{s} * 0{,}25\ s = 0.335\ m$$

Possible discrete distance are $\frac{2}{8}m = 0{,}250\ m$ or $\frac{3}{8}m = 0{,}375\ m$

To minimize the error we chose the value $0{,}375\ m$.

Herewith the resultant absolute error is

$0{,}375\ \mathrm{m} - 0{,}335\ \mathrm{m} = 0{,}04\ \mathrm{m}$ or a relative error from

$$\frac{0{,}04\ m}{0{,}335\ m} * 100 \approx 12\ \%$$

Moving: In the last step the precalculated shifts for each student are performed. This happens with the no crossing path method. This technique allows each student to move on its precalculated way until he has performed all shifts or crossed a trajectory of another individual who has already moved.

7 Elevators

An important part of each house, which has more than a few floors, are the Elevators. This element of the simulation is needed because without elevators the simulation is not really complete and indispensable for the simulation of persons with physical handicaps. To control the activity of the elevators an event-based approach was used. An elevator has seven distinct states (wait, move up, move down, open door, close door, person get in, person get out). Each of them serves exactly one purpose during operation (no superstates).

After an event has completed, the controller selects a new task depending on which car and floor buttons have been pressed. A flow diagram which shows all possible transitions from one state to another is displayed in the figure below.

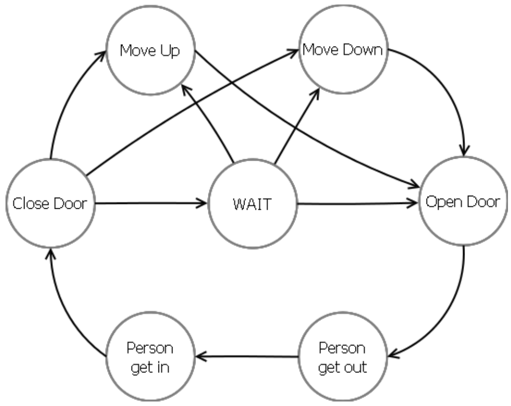


Figure 6. Possible elevator transitions.

In the course of implementation, we have found that the control of clustered elevators is not as easy as it seems at the beginning, since it is not always clear which elevator must be directed to which floor. For example, a floor call ordered an elevator in his floor. Now it possible that one elevator passes this floor anyway while another elevator is waiting at the ground-floor.

Now, shall the second elevator process the call or should the first elevator interrupt his ride to pick up the waiting passengers? This is just one question for the developer of the elevator controller. In order to keep this part of the simulation easy a relative simple algorithm of a controller is in use. He has only two major rules:

- If an elevator is waiting on a floor, send him to the waiting passengers.
- If no waiting elevator is available stop the next cabin that passes the floor.

With those two points the control of the elevator works surely not in the best an efficient way but the algorithm is easy enough to implement and adequate for our needs.

The next big problem witch the elevators indirectly creates was that each person in the simulation strives to take the shortest and supposedly the best path through the building. We have defined the length of paths between floors when using elevators as zero. As a matter of fact, the shortest path through the building is always by the way of the elevators, but this does not have to be the fastest way. To handle this problem a combination of shortest and fastest path is used. To calculate the time via the elevators each person who used one informs the controller after his ride about the waiting time. With the aid of this data that is stored in files for following simulation runs an estimation can be calculated to give the persons a pointer for the approximate waiting time.

A special potential hazard for collisions between people and following deadlocks are the entrances of the elevators. To disarm this each person, who wants to take the elevator, registers him- or herself at the end (Remark: Persons with physical handicap at the beginning) of a waiting list, while situating themselves near the elevator area. They don´t enter this area which is represented in the picture with the dark blue color.

When the cabin arrives the floor the first step is to exit all passengers, which will be inserted into this empty, for this purpose reserved, elevator area with a specified delay between each person. The second step is to remove the waiting passengers, as many as possible, depending on the capacity of the cabin, with the same delay time, from the discrete grid and mark them as “now in the elevator”. With this method the movements in front of the elevator are reduced to a minimum and so is the possibility of a deadlock.

8 Physical Handicapped People

To consider the special needs of people with physical handicaps this simulator also contains the possibility to generate people which rely on a wheel chair. It exists an adjustment for the probability that a random generated person is physical handicapped. The major different attributes of these persons are:

- The need of a special speed distribution.
- An increased required space. Instead of 4 x 4 cells wheel chairs need a space of 6 x 6 cells.
- As opposed to persons with no physical handicap wheel chair users do obviously not have the possibility to reduce their used space.
- The priority in the elevator waiting queues. This is not necessary for the simulation but in the real world commonly used.
- The inability of using stairs or overcoming other obstacles. To simplify the model it will not be considered how small these obstacles are. That means that already one stair, which can be handled in the real world by many of wheel chair users, is in the simulation an insurmountable obstacle.

The last point needs a little more effort in the implementation into the software. To solve this problem an independent routing algorithm is implemented, which works basically in the same way as the original with the one exception, that stairs are handled in same way as walls.

This additional algorithm implicates also that for each cellular automaton a second static floor field is used, doubling the amount of memory.

9 Conclusion

Combining different model types to implement hybrid simulation systems can solve some classical problems, but some other problems may occur. Some advantages were mentioned. CAs were introduced in the simulation system to represent a model computing vacation times. A standard software for implementing parts like queuing servers would be a huge amount of work.

Some disadvantages like implementing interfaces for connecting the different “sub models” had to be accepted. The work on this project is in progress, the quantitative results for the whole system with about 9.000 rooms and - in the future - maybe about 30.000 students has to be validated –for a <morespace> overview see [8].

References

[1] A. Kirchner, H. Klüfel, K. Nishinari, A. Schadschneider, M. Schreckenber, *Discretization effects and the influence of walking speed in cellular automata models for pedestrian dynamics.*

[2] F. Weifeng, Y. Lizhng, F. Weicheng. *Simulation of bi-direction pedestrian movement using a cellular automata model.* Physica A 2003.

[3] M. Bruckner. *Modellierung der Fußgängerdynamik im universitären Betriebmit Hilfe von Zellulären Automaten in der Programmiersprache JAVA. Vienna University of Technology 2009*

[4] P. Molná., *Modellierung und Simulation der Dynamik von Fußgängerströmen. Shaker Verlag, Aachen 1996.*

[5] T. Fujiyama, N. Tyler. *An Explicit Study on Wallking Speeds of Pedestrians on Stairs.* Centre for Transport Studies. University College London.

[6] T. Kretz, A. Grünebohm, M. Kaufman, F. Mazur, M. Schreckenberg, *Experimental study of pedestrian counterflow in a corridor.* Journal of Statistical Mechanics: Theory and Experiment. 2006

[7] V. J. Blue, J. L. Adler, *Cellular automata microsimulation for modeling bi-directional pedestrian walkways.* Transporation Research Part B 2001

[8] S. Tauböck, F. Breitenecker, D. Wiegand, N. Popper. *The <morespace> Project: Modelling and Simulation of Room Management and Schedule Planning at University by Combining DEVS and Agent-based Approaches.* SNE Simulation Notes Europe 21(1), 11-20, ARGESIM Vienna 2011. doi: 10.11128/sne.tn.21.10039

Submitted: June 2012
Accepted: August 1, 2012

The Modelica Library 'AlternativeVehicles' for Vehicle System Simulation

Thomas Braig[1], Holger Dittus[1 *], Jörg Weiss-Ungethüm[2], Tobias Engelhardt[1]

[1] German Aerospace Center (DLR), Institute of Vehicle Concepts, Pfaffenwaldring 38-40, 70569 Stuttgart, Germany
[2] German Aerospace Center (DLR), Institute of Technical Thermodynamics; **holger.dittus@dlr.de*

Abstract. The AlternativeVehicles Library (AV) allows calculating the energy demand and optimizing the energy management for conventional and alternative vehicle concepts. The components provided focus on the simulation of alternative power trains. Therefore models of energy storages and energy converters such as electric drives and fuel cells are included.

The library was developed within the European research project Eurosyslib. The current version includes contributions of the DLR Institute of Robotics and Mechatronics and is distributed by the Bausch-Gall GmbH. Within this paper architecture and components of the AlternativeVehicles library are described. Exemplarily fuel consumptions of a conventional vehicle and a parallel hybrid vehicle are compared.

Introduction

The activities at the Institute of Vehicle Concepts contribute to the sustainable development of technological systems for future generations of road and railway vehicles. Therefore components for energy conversion and energy recuperation are developed and integrated into research vehicles [1]. Also the scenario tool Vector21 has been developed, which allows assessing the influence of car specific fuel consumption, energy prices, taxation, customer decision etc. on the future vehicle fleet and total CO2 emissions [2]. System level simulations of different vehicle concepts are essential for all of these activities. Due to the lack of commercially available, flexible, appropriate tools new models have been created in Modelica.

During the last years the Modelica library AlternativeVehicles (AV) has been developed with contributions of the DLR Institute of Robotics and Mechatronics and since March 2011 it is commercially available.

1 Overview of the AlternativeVehicles Library

The AlternativeVehicle Library includes different vehicle architectures i.e. a conventional ICE powered vehicle, an electric vehicle with combustion engine based range extender, a pure electric vehicle with a high-voltage battery as energy storage, a parallel hybrid vehicle, and a fuel cell hybrid electric vehicle. To enable fast simulations of entire vehicle systems, mainly concentrated modeling approaches are used for the component models. The components provided focus on the simulation of alternative power trains. Therefore models of energy storages (battery, double layer capacitor) and energy converters such as fuel cells, electric motors and power electronics are included.

In addition to the component models the library includes various parameter sets for component models representing technical data of real life components. Some component models are available in different detail levels which enables the user to choose an appropriate modeling approach depending on the objective of the simulations. Additional vehicle concepts can be easily created by combining the available components and developing the control strategy.

The AV is modeled in Modelica. Modelica is a simulation language which is non-proprietary, object-oriented and equation based. It allows to model complex multi-physical systems containing, e.g., mechanical, electrical, electronic, hydraulic, thermal, control, electric power or process-oriented subcomponents [3]. The models contain interfaces and mathematical equations (algebraic or ordinary differential equations). Physical interfaces, such as heat ports, electrical pins or mechanical flanges include flow and potential variables.

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 22(2), 2012, 101-106 | doi: 10.11128/sne.22.tn.10133

Modelica models do not only describe the mathematical behavior of the real world system but also have the same structure. This leads to intuitively creation and understanding of the models.

To ensure the compatibility to other automotive Modelica libraries, the AV is based on the freely available VehicleInterfaces library [4]. The components are connected via different interfaces: physical connectors and signal buses (e. g. Figure : flange, signal bus).

2 Vehicle Architectures: Conventional Vehicle and Parallel Hybrid

Several vehicle architectures are provided within the AV. In Figure 2 and Figure 1 the top level models of a conventional vehicle and a parallel hybrid vehicle are shown.

The conventional vehicle consists of the following top level components: driver, control module, accessories, engine (ICE), transmission, driveline, chassis and inspector. The parallel hybrid vehicle has additionally an electric motor, a high voltage battery, and a clutch between the engine and the electric motor [5].

The control module of the parallel hybrid is labeled hybrid control unit (HCU). Every top level component is connected with the control module and has its own status and control sub bus, e.g. EngineStatusBus and EngineControlBus.

The status buses contain signals describing the actual state of the system and the control buses contain signals to control the system such as setpoint values. The top level components of the parallel hybrid vehicle are:

Driver: The "StandardCycleDriver-Manual" is a driver model with gearshift and clutch control suitable for ICE driven vehicles with manual gearshift. The core of the "StandardCycleDriverManual" model is a PI-controller which is fed by the set-point speed from the driving cycle. Clutch and gearshift control is done by several embedded submodels. The driver model includes various driving cycles (NEDC, HYZEM, FTP75, Artemis etc.).

ControlModule: The controlModule (hybrid control unit, HCU) controls the other top level components. All settings related to the operating strategy are made in the HCU. It enables simulation of several kinds of hybrids with different operating strategies.

Engine: The engine is a table-based model of an internal combustion engine (ICE). Torque and fuel consumption are defined by tables, which can be adapted to the desired engine. The model optionally includes an idle speed controller, an overdrive protection and a starter motor.

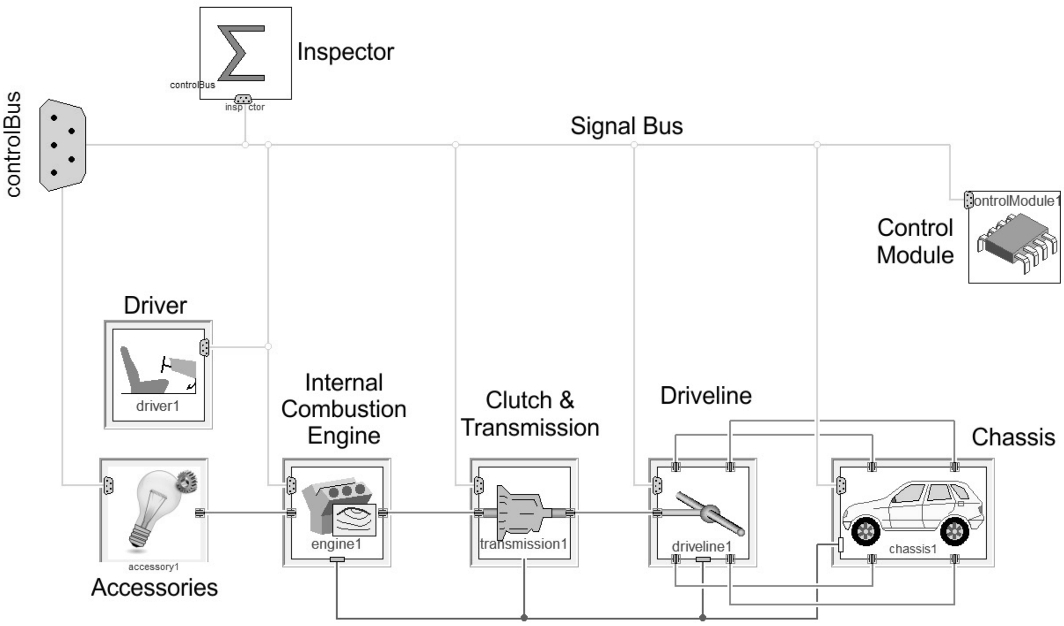


Figure 1. Top level model of the parallel hybrid architecture including internal combustion engine (ICE), electric motor with additional clutch, battery, electric accessories and the hybrid control unit (HCU). The HCU controls the functions of the hybrid system, i.e. start/stop of the ICE, load point shifting, state of charge (SOC) and brake energy recuperation.

Electric Motor: The “SimpleDrive4 Tab” is a table-based model containing 4 tables: Torque and efficiency for both directions (motor, generator). The EM itself can be overloaded but the HCU is not designed to use this option.

Clutch 1: The clutch 1, which is controlled by the HCU, can optionally be disabled (e. g. in case of a micro hybrid)

Transmission: The “SimpleManualGear” is a model of a manual transmission. It is a combination of clutch and gearbox

Driveline: The driveline model is of a rear-wheel drive 4-wheeled vehicle. The rear differential is modeled using an ideal gear and planetary gear.

Chassis: The “TwoWheelPolynomial” is a single tracked (two wheels) model with polynomial driving resistance.

Battery: Impedance based battery model using 2-dim parameter table lookup.

Accessories: Electric driven auxiliaries represent any electric load of a vehicle. In conventional vehicles are servo motors, fans, etc. the major electric loads. In full electric vehicles also the AC system is electric. This model provides just a constant electric load.

Inspector: The inspector is used for post-processing. It calculates characteristic numbers as efficiencies or fuel consumption.

3 Hybrid Control Unit (HCU)

The task of the hybrid control unit (HCU) is controlling the complex hybrid drivetrain. It contains the operating modes:

Electric Driving: This mode is used in vehicles where the electric drive torque is sufficient to power the vehicle exclusively. The ICE is turned off and the vehicle is moved by the EM like an EV.

Start-Stop: The ICE is shut down whenever the vehicle stops and is turned on again when the vehicle starts. Since the EM is more powerful than a normal starter, the ICE starts more quickly and with less noise than in conventional vehicles.

Load Point Shifting (LPS): ICEs and especially gasoline ICEs have poor efficiency at low loads. In low load situations the EM generates electricity, charges the energy storage (e.g. battery) and rises the demanded load of the ICE.

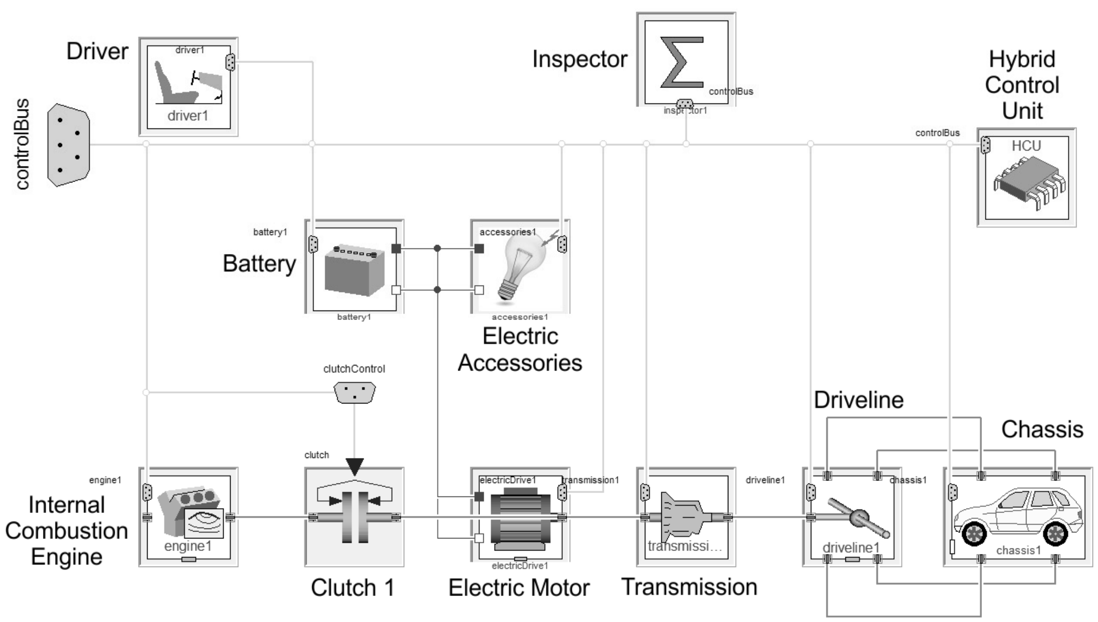


Figure 2. Top level model of the conventional vehicle architecture including the main components internal combustion engine (ICE), accessories, transmission, driveline and chassis. The driver model allows choosing from different driving cycle such as NEDC, Artemis, FTP75. The control module model contains the control units for accessories, transmission and internal combustion engine.

Recuperation: Recuperation or regenerative braking means the recovery of kinetic energy by the EM working as a generator to charge the battery.

Electric Boost: The EM assists the ICE to accelerate the vehicle. The EM has great torque at low angular velocity where the ICE has a weak spot. The combination of both results in a higher and more constant torque distribution. The electric boost can either occur to start the vehicle (low acceleration, startup assistance) or when high acceleration is required.

The driver is no longer in direct control of the way the vehicle delivers propulsion. In contrast to a conventional vehicle, where the gas pedal signal of the driver is passed through unchanged to the ICE, the gas pedal signal in a HEV only indicates the desired amount of propulsion. The HCU calculates how to deliver the required propulsion most efficiently. In order to do this, it uses an implemented algorithm which depends on many variables and parameters (operating strategy). The HEV is able to perform various operating modes. All operating modes depend on various conditions.

The operating strategy is crucial for the efficiency and thus the fuel consumption of the vehicle. The HCU needs to calculate the most efficient operating mode at any time always considering the state of all components (e.g. maximal available engine torque, SOC of the energy storage).

The HCU consist of subunits, signal processing, sub buses of the top level components and sub buses of the subunits. Some of the above mentioned operating modes are implemented in a special subunit (e.g. Start_Stop, Figure). The Start_Stop (controller) sub unit is responsible for stopping and starting the ICE. It has three output signals: A start and a stop signal for the ICE and, since there is no starter included in the ICE, a controlLever signal for the EM. The ICE starts if it gets an engineStart signal and it is simultaneously accelerated by the EM. For shutting down, the engineStop signal is sufficient. Starting and stopping the ICE occurs in a certain order. In the model this order is maintained using the state graph library of the modelica standard library (Figure).

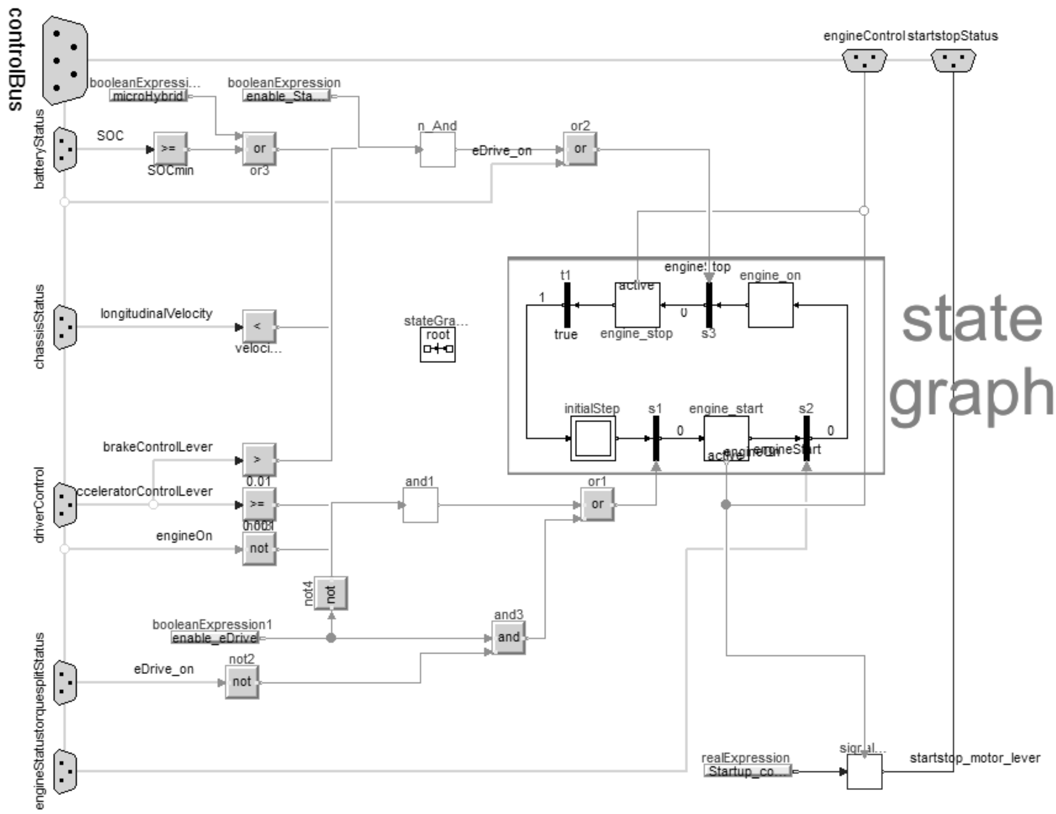


Figure 3. Start-Stop subunit of the hybrid control unit. This subunit controls switching off and starting up the internal combustion engine depending on model signals such as SOC, velocity and torque request.

4 Simulation Models and Results

In the AV parameterized models are available, e. g. a VW Golf V conventional vehicle, a DLR fuel cell hybrid electric vehicle and a DLR battery electric vehicle. The Mercedes-Benz S 400 BlueHYBRID (S 400 H) is a parallel hybrid without clutch between ICE and EM (Figure , clutch 1 closed). In this car the operation modes Start-Stop, regenerative braking, and electric boost are available. The S 400 H is based on the Mercedes-Benz S 350. Both vehicles are parameterized and simulated for comparison [5].

Both vehicles are driven by the M 272 KE engine which is a naturally aspirated V6 gasoline ICE with intake-manifold fuel injection. They also share the NAG2 automatic 7-speed gearbox. The EM is located between ICE and gearbox. It delivers 15 kW as motor and 19 kW as generator.

The Lithium-Ion battery of the vehicle contains 0.8 kWh and supplies the power electronics and also the electric AC compressor. A DC-DC converter is located between the 126 V high-voltage battery and the 12 V battery. The vehicle has rear wheel drive. All parameterization data are taken from literature [7][8].

Figure 5 shows the longitudinal velocity of the NEDC, the state of charge (SOC) of the battery and the EM-controlLever, which is the name of the signal variable used to control the electric torque. The end value of the SOC equals the start value. The EM-controlLever value becomes positive when the vehicle is accelerating (electric boost, startup assistance) and negative when the vehicle is braking (recuperation). A positive EM-controlLever value causes a decrease of the SOC and a negative EM-controlLever value causes an increase of the SOC. If the EM-controlLever equals zero, the SOC is slightly decreasing due to the accessories (120 W).

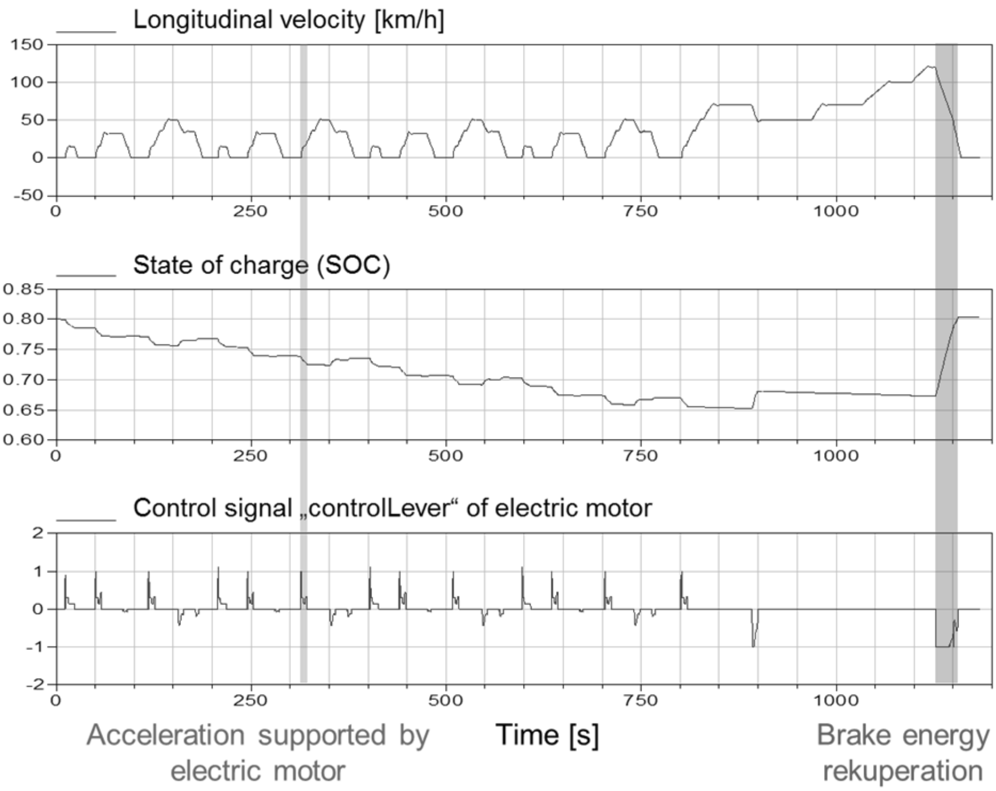


Figure 5. Simulation result plots of vehicle velocity, state of charge and electric motor controlLever. During accelerations the electric motor supports the combustion engine and the state of charge of the battery decreases. During braking the electric motor works as a generator and recuperates kinetic energy from the wheels, leading to an increase of the battery's state of charge.

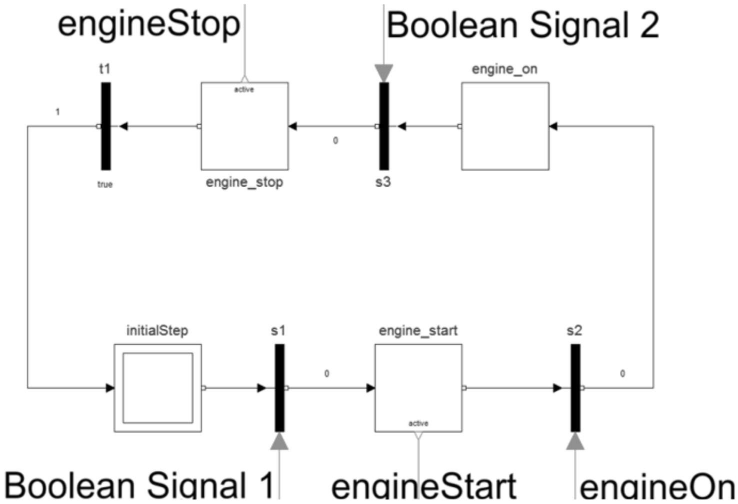


Figure 4. Assignment of engineStart and engineStop signals with state graphs.

Abbreviations

AV	AlternativeVehicles Library
DLR	Deutsches Zentrum für Luft- und Raumfahrt e. V. (German Aerospace Center)
EM	Electric Motor
EV	Electric Vehicle
HCU	Hybrid Control Unit
ICE	Internal Combustion Engine
LPS	Load Point Shifting
NEDC	New European Driving Cycle
SU	sub unit (of HCU)

For the S 350 the result is 10.0 l/100km which is very close to the fuel consumption specification of the S 350 (10.1 – 10.3 l/100km [8]). The average value of the S 400 H is 8.56 l/100km. This result differs from the fuel consumption specification of the S 400 H (7.9 – 8.1 l/100km [8]) by 5 %. Several reasons might lead to this overestimation, e. g. deviations in the driving resistances or the accessories are overestimated.

5 Conclusion

The AV is a commercially available, open source automotive Modelica library which allows the user to model various vehicle concepts, develop operating strategies and calculate the energy demand for different driving cycles. The main contents of the AV are models and parameterized examples for several alternative vehicle architectures, energy storages and energy converters.

In the paper the models and simulation results of a conventional vehicle and a parallel hybrid vehicle have been compared. The calculated fuel consumptions fit the values of the specifications sheet quite well. The deviations can be caused by several reasons, e. g. deviations in the driving resistance or overestimated accessories.

References

[1] T. Braig, J. Ungethüm. *System-Level Modeling of an ICE-powered Vehicle with Thermoelectric Waste-Heat-Utilization*. In *Modelica Conference Proceedings*, Como/Italy, 2009.

[2] P. Mock, D. Hülsebusch, J. Ungethüm, S.A. Schmid. *Electric vehicles – A model based assessment of future market prospects and environmental impacts*. In Proceedings of 24th Electric Vehicle Symposium. Stavanger, Norway, May 13-16, 2009.

[3] Modelica Association: *Modelica and the Modelica Association*. https://www.modelica.org/.

[4] Dynasim, DLR, and Modelon: *VehicleInterfaces Library*. Version 1.1. Tutorial and Reference, 2008.

[5] T. Engelhardt. *Modeling and Simulation of a Parallel Hybrid Vehicle in Modelica.* Studienarbeit (unpubl.), 2011.

[6] J. Ungethüm, D. Hülsebusch, H. Dittus, T. Braig. *Simulation von alternativen Fahrzeugantrieben in Modelica.* In ASIM Conference. Ulm 2010.

[7] P. Hofmann. *Hybridfahrzeuge*. Springer-Verlag, Wien, 2010.

[8] Vollrath, Armstrong, Schenk, Bitsche, Lamm. *S 400 BlueHYBRID – First Hybird Vehicle with Li-Ion-Technology*. In: 30. Internationales Wiener Motorensymposium, 2009

Submitted: September 2011
Revised: July, 2012
Accepted: August 1, 2012

Flexible Task Oriented Robot Controls Using the System Entity Structure and Model Base Approach

T. Schwatinski*, T. Pawletta, S. Pawletta

Wismar University of Applied Sciences, Dept. of Mechanical and Process Engineering,
Res. Group Computational Engineering and Automation (CEA),
PB 1210, 23952 Wismar, Germany; **tobias.schwatinski@hs-wismar.de*

Abstract. A new method used for developing Flexible Task-oriented robot Controls (FTC) using the System Entity Structure (SES) is introduced. Task-oriented robot controls are based on the composition of atomic tasks aimed at achieving a previously specified goal. Flexible task-oriented controls differ in that the composition of atomic tasks is not fixedly predefined but is composed during the operation of the control based on actual process states and with respect to constraints in the sequence of tasks. The System Entity Structure is an ontology, which can be used for hierarchically representing system compositions. For cooperating robots the paper shows how to generate and execute FTCs specified by an SES and an associated model base (MB).

Introduction

Flexible Task oriented robot Controls (FTC) consists of several atomic tasks that are composed with respect to any constraints of their sequence with the aim of achieving a specific goal. The concrete sequence of atomic tasks can be determined either on the basis of actual process states during control operation or based on predictive process simulations. Therefore, FTCs belong [1] to intelligent robot controls and are related according to their implementation, due to requirements and complexity, to “large-scale” development [2]. As a consequence implementing such robot controls has to follow a systematic development process.

This paper presents the FTC/SES method used for systematically developing Flexible Task-oriented robot Controls (FTC) based on the System Entity Structure (SES) and Model Base (MB) formalism. The SES is a basic element of the FTC/SES method. The SES was originally developed in the eighties by Zeigler [3] and has been enhanced continuously to data engineering [4] until today.

The SES is an ontology designed for the hierarchical representation of real or imagined systems and is mostly used for defining meta models in the field of simulation technique. In our research the SES is used for specifying flexible industrial robot controls including subordinated process components. Using the SES the overall control task is divided into subtasks that are composed of atomic tasks and other composed subtasks. The declarative and modular, hierarchical specification of a control, including its process components using a SES, enables systematic control development. It also supports its reusability, adaption and maintenance.

Additionally, the FTC/SES method is based on the Simulation-Based Control (SBC) approach [5] and supports the successive development of simulation models within a homogeneous computing environment, beginning from the design phase until the operation phase. Below, the basics of the SES and the SBC are introduced in brief. Next using an example, their combined usage for specifying robot controls is discussed. Subsequently, the automatic generation of executable controls is shown. Finally, the paper summarizes important aspects of a prototype implementation and some experiences are summarized.

1 System Entity Structure and Simulation-Based Control Approach

1.1 The System Entity Structure

The System Entity Structure (SES) is an ontology. The SES forms a tree, the nodes of which can be categorized [4] as four node types: (i) entity, (ii) aspect, (iii) multiple-aspect and (iv) specialization. The general sequence of nodes in an SES is shown in Figure 1 (a). Entity nodes represent elements of the real or imagined world. Aspect nodes are used for decomposing entity nodes into finer-grained structures.

SNE Simulation Notes Europe – Print ISSN 2305-9974 | Online ISSN 2306-0271
SNE 22(2), 2012, 107-114 | doi: 10.11128/sne.22.tn.10135

Multiple-aspect nodes define multiplicity of entity nodes and specialization nodes represent categories or families of characteristics of entities. In addition attributes and their domain of definition can be attached to any node. Figure 1 (b) shows an example of an SES for specifying several automotive types. Every entity car consists according to the aspect node car of the entities engine, wheels and chassis. The entity engine is either specialized to the entity diesel or the entity gasoline and an attribute engine capacity is inherit to both. Moreover, the attribute stroke_cycle has been attached to the entity gasoline, which can be set to the values two or four. The entity 'wheels' is followed by a multiple-aspect node with a multiplicity of 4. Hence, this node will break down into four entities of type winter tire.

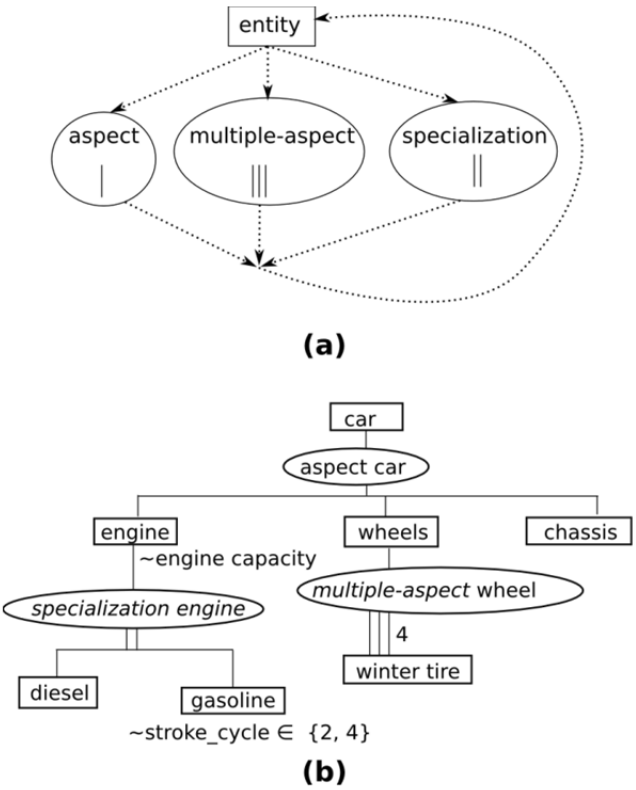


Figure 1. General structure of an SES.

Therefore the SES in figure 1 (b) characterizes the three different automotive types:

- diesel engine with a specific engine capacity, four winter wheels, chassis
- two-stroke cycle gasoline engine with a specific engine capacity, four winter wheels, chassis
- four-stroke cycle gasoline engine with a specific engine capacity, four winter wheels, chassis

In doing so, the SES can be used for clearly defining different characteristics of any composite system. All coupling relations between the entities have to be specified at aspect nodes. Moreover, the SES supports the specification of selection constraints. This means that the selection of an entity in a specialization may cause the selection of a certain entity in another specialization.

The simple example in Figure 1 does not define any constraints. Originally the SES was developed for specifying models in simulation technique field. The SES in combination with a model base (MB) that contains an executable software component for each leaf node of the SES allows simulation models to be generated automatically.

1.1 The Simulation-Based Control Approach

The Simulation-Based Control (SBC) approach described in [5] is a specific type of the software in the Loop (SiL) principle [6] and supports the usage of simulation models throughout during the entire development process of controls. Simulation models are enhanced stepwise beginning from the design phase to the automation phase and finally are used as control software directly during the operation phase using implicit code generation.

The control software (program) is executed using a real-time simulator. This approach means a development PC or industrial PC can be used to control real processes. The entire development process of controls based on the SBC approach is shown in Figure 2.

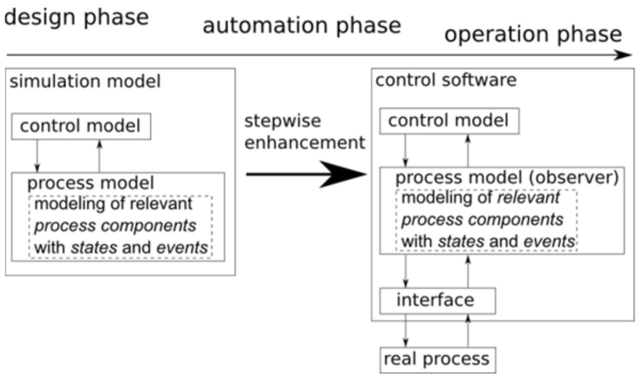


Figure 2. Simulation-Based Control (SBC) approach.

The SBC approach defines that any control software consists of a control model, an interface and, if required, a process model. The interface provides the connection between the real process and the control software.

This specific type of communication with a real process is called implicit code generation. The process model maps the behavior and states of real process components. In addition the process model that has already been used in the automation phase can be integrated into the control software as a state observer.

This procedure can increase the quality of controls, e.g. by calculating additional or immeasur-able state values. The control model maps the complete control logic.

2 Developing Robot Controls based on a Declarative Specification

2.1 Integrating the SES/MB Formalism and the SBC Approach

The SBC approach supports effective implementation of robot controls from the beginning of the design phase to the end of the operation phase using simulation models. The SES/MB formalism supports a systematic and declarative specification of dynamic systems by means of a tree structure (SES) and automatic program generation using predefined param-eterizable modules from a model base (MB). In the following both approaches are used for defining task-oriented robot controls and it will be shown how highly flexible controls can be implemented.

The SBC approach supports following [5] the definition of task oriented controls. Predefined atomic tasks are composed and parameterized within a control model according to a specific control objective. In doing so any control commands and any reactions on system states are programmed in atomic tasks. This basic principle is shown schematically based on a simple control model in Figure 3. It shows that the sequence of atomic tasks is fixed within a control model. The whole flexibility of a control has to be implemented inside the atomic tasks and by means of their coupling relations. In particular complex and flexible robot applications comprise multifaceted relations between the atomic tasks within a control model as well as between the control model and the process model as a whole. This leads to highly complex controls because the SBC approach supports a structure-oriented modeling but unfortunately supports only a fixed composition of tasks.

The declarative specification of controls and the process-dependent generation of executable controls using the SES/MB formalism counteracts this drawback of the SBC approach.

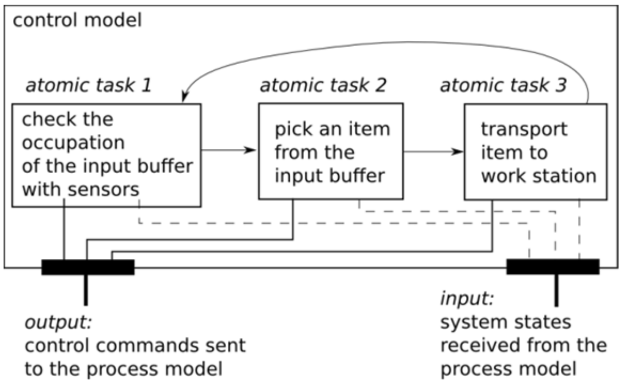


Figure 3. Coupling of atomic tasks in a control model following the SBC approach.

Integrating the SES/MB formalism and the SBC approach for implementing Flexible Task-based Robot Controls is called the FTC/SES method. The major ideas behind this method are specifying a flexible control strategy with an SES and successively generating temporary controls during process operation. The most important elements and interactions of a flexible task-oriented robot control following the FTC/SES method are pictured in Figure 4. It is based on an adaptive control approach, which consists of a monitoring, a decision and a control generation and execution component. The monitoring component monitors every change in system states and the occurrence of control events during the execution of the temporary current control and continuously passes significant information to the decision component.

Based on the information received by a decision maker, the decision component checks whether the temporary control has to be adapted. If adaption is necessary, a new control structure is derived by analyzing the SES and afterwards passed to the control generation and execution component. The control generation and execution component generates and executes temporary control variants. A (model) generator creates a new temporary control. It evaluates the received control structure and builds the new control program using the predefined components in the model base.

According to the SBC approach, the control program consists of a control model C, a process model P and a process interface I and is executed by means of an event-oriented simulator that acts in real-time. The generated control program may be of limited lifetime or may be interrupted in consequence of real process events.

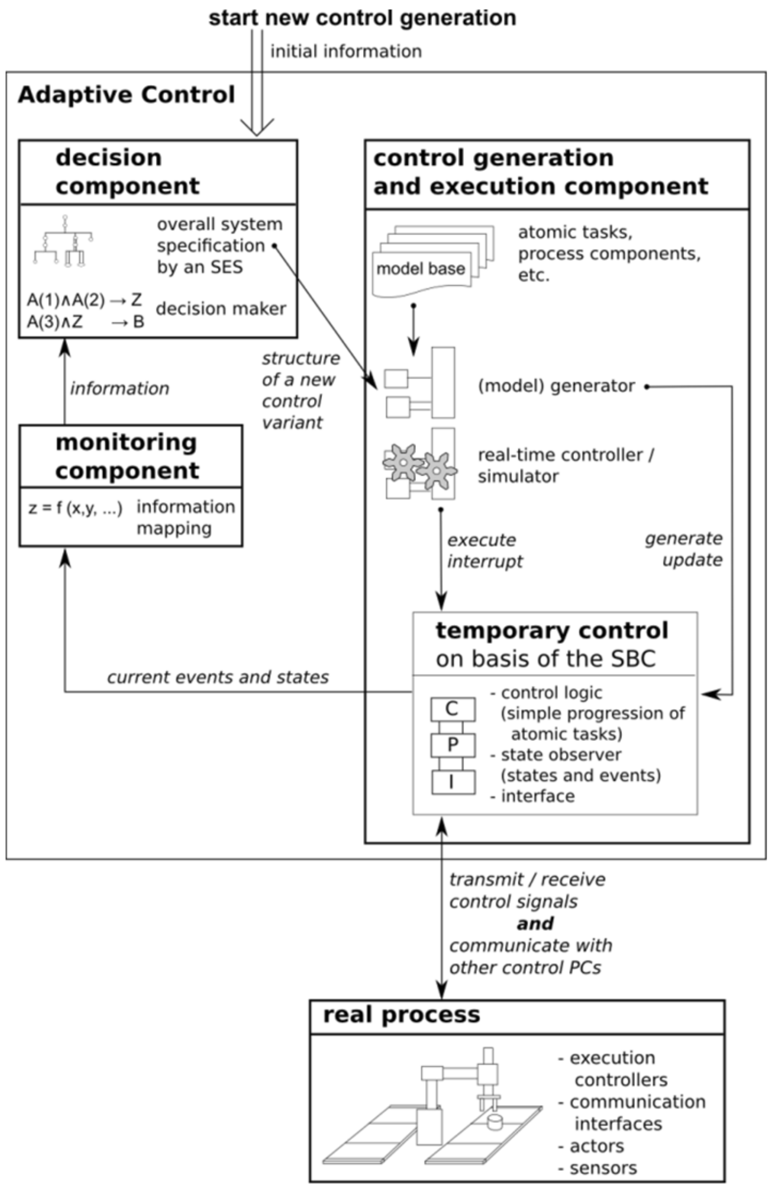


Figure 4. Elements and interactions of a flexible task oriented robot control following the FTC/SES method.

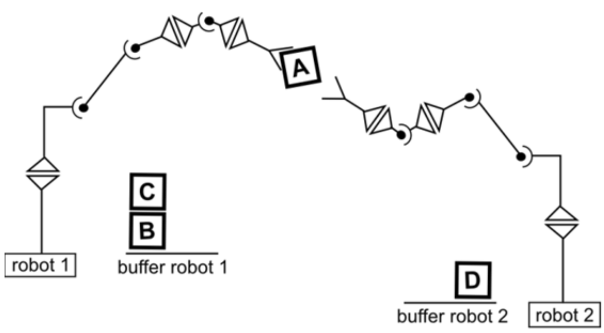


Figure 5. Cooperating robot application with two robots and two separated buffers.

In these cases a new control program is generated according to the described steps. This procedure is repeated until any predefined abort criteria occur. The declarative specification of robot controls and automatic generation of temporary control variants are discussed in detail in the following subsections.

2.2 Declarative Specification of Robot Controls

In this research we propose a slightly modified SES formalism called control-SES for specifying flexible industrial robot controls. The fundamental ideas are discussed using a small application shown in Figure 5. The application consists of two cooperating robots, each of which has a separate buffer.

The objective target for both robots is to re-arrange the objects in both buffers according to a user defined order. To fulfill this objective the robots have to cooperate, because objects are stacked in the buffers. The total amount of places in the storage areas is much smaller than the total amount of objects. Each robot has to cache objects from the other robot in its own buffer so that the other one can operate its necessary sort sequence. The transfer of objects takes place directly between both robots. At the beginning the control has to determine an optimal sort sequence to minimize the total amount of steps.

A simplified control-SES of the described robot application is shown in Figure 6. The pictured control-SES consists of two parts.

The upper part specifies the time invariant part of the control that according to the SBC approach defines a control model, a process model and a process interface. The overall task of the robot control is structured in several smaller control tasks and process interactions which are specified in the lower part. This part of the control-SES presents the time variant characteristics in terms of specialization nodes, which are used to specify the alternative use of atomic tasks in the control model and interactions between robots and buffers in the process model.

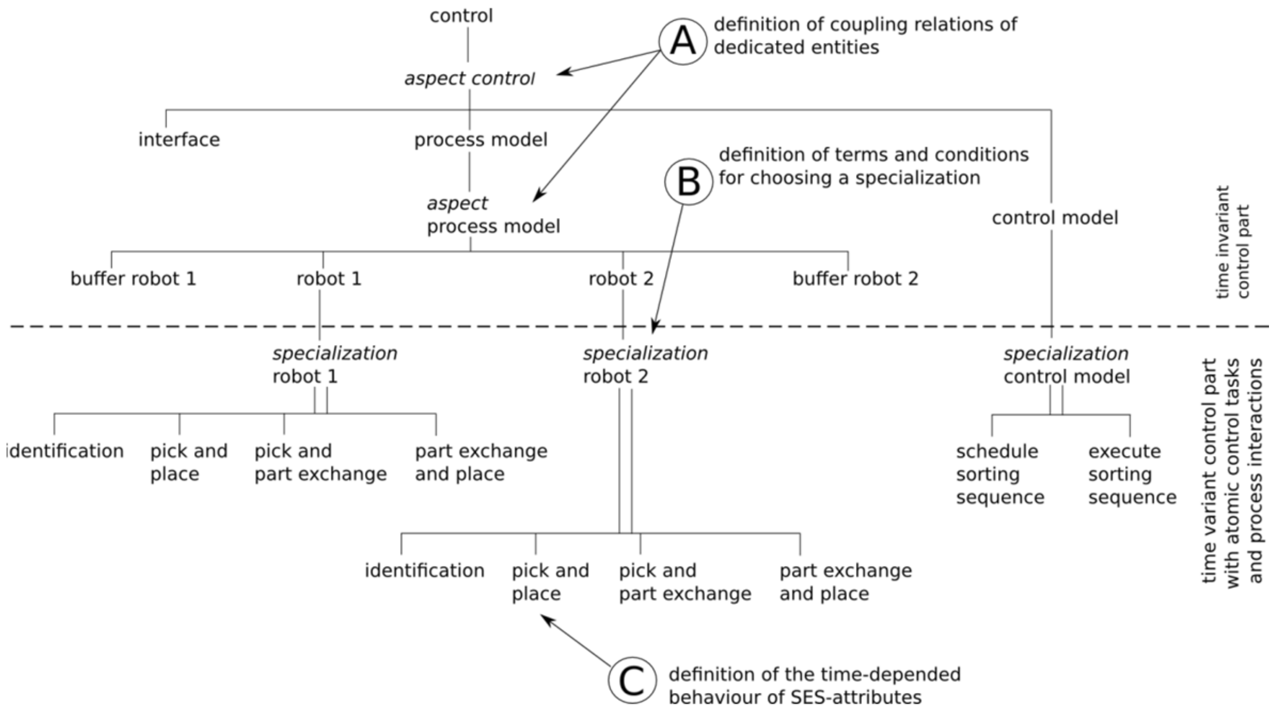


Figure 6. Declarative specification of a cooperative robot control using a control-SES.

The leaf nodes (C) of the control-SES are no further decomposable entities which are implemented as executable software components and stored in a model base (MB). The control-SES in Figure 6 is incomplete to preserve clarity. Beside nodes and edges a control-SES specifies node attributes. The leaf nodes representing atomic control tasks and process interactions define the modification of these attributes depending on the real process behavior. These attributes are described in more detail in the next subsection. The aspect nodes (A) define the coupling relations of the succeeding entities. Furthermore, the specialization nodes (B) define selection rules used for choosing dedicated atomic tasks and process interactions.

The general structure of all possible control variants follows from the time-invariant, upper part of the control-SES. A valid control variant is synthesized from the control-SES using a parameter vector that maps the current process behavior in terms of states and events to attributes of the control-SES. The result of this synthesis is a reduced tree structure in which all decision nodes like specialization or multiple-aspect nodes are resolved. Following [4] this procedure is called “pruning”.

During the pruning process all entity nodes in the undermost layer of the time invariant part of the control-SES are substituted with leaf nodes of the time variant part. Atomic control tasks and process interactions are selected in the specialization nodes. The structure of two temporary control variants synthesized from the control-SES pictured in Figure 6 is shown in Figure 7. The control structures pictured in Figure 7 represent only a subset of all valid temporary control variants according to the described application.

2.3 Sequence of Atomic Tasks and Process Interactions

The sequence of atomic control tasks and process interactions is determined during control operation on the basis of current process states and events. The flexibility of the control follows from the iterative synthesis and generation of temporary valid control variants. Therefore, the decomposition of the entire control in appropriate atomic control tasks and process interactions is a prerequisite. Specifying process interactions and defining their sequence is shown by robot 1. For this purpose, Figure 8 shows the description of entity node *robot 1* and its succeeding nodes in more detail.

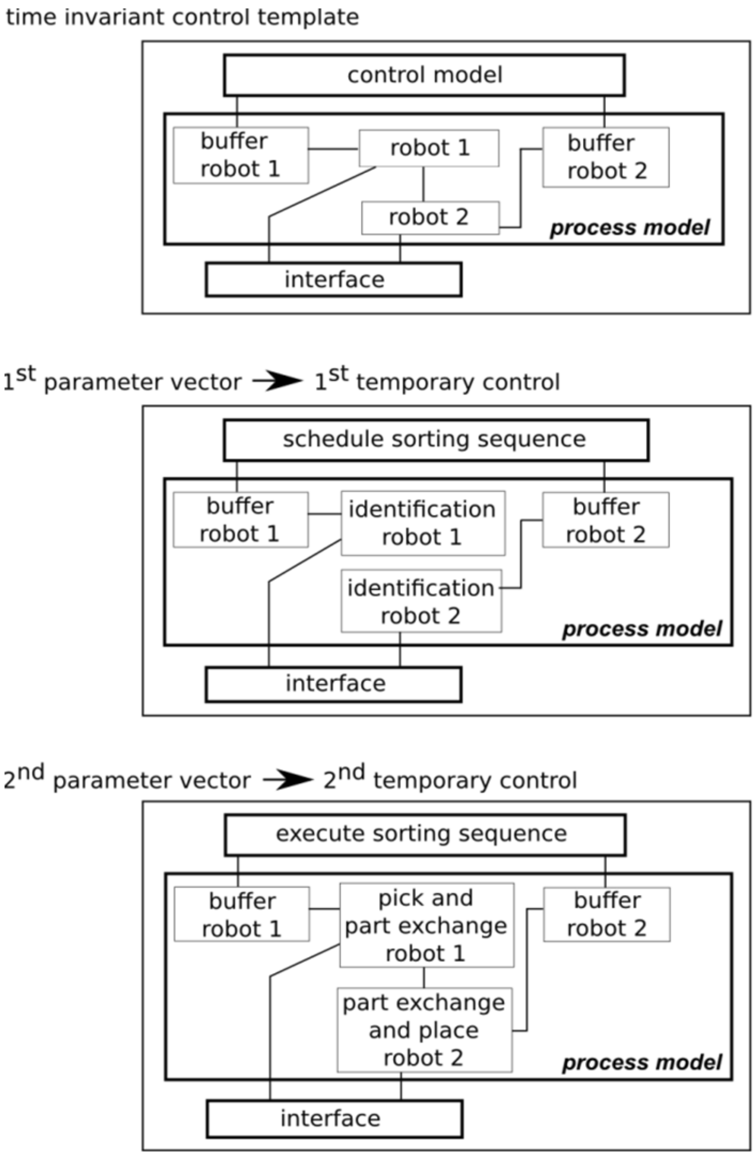


Figure 7. Generation of temporary control variants from a control-SES.

Inheritable parameters, including their domain of definition, are defined with attributes at entity node *robot 1*. In this example *roadmap* is a robot-specific set of points and connections in the configuration space of the robot used for path planning. The succeeding node *specialization robot 1* defines selection rules for process interactions evaluated in the control synthesis phase.

Every atomic control task or process interaction can define a time, state and event dependable behavior of SES-attributes using the operator “?”.

If, for example, the interaction *identification* in Figure 8 is part of the current control, the SES-attribute *interactionRobot1* is declared changeable during control operation by the expression “interactionRobot1 = ?”. In addition, the PRUNE action defines a new control synthesis if the value of *interactionRobot1* is set to one element of the set {*pickPlace, pickExchange, exchangePlace*}.

If, for example, a *pick and part exchange* interaction should follow after an *identification* interaction, then the SES-attribute *interactionRobot1* is set to *pickExchange* and a PRUNE action has to be performed.

That means a new control structure has to be derived by re-analyzing the control-SES using the modified SES-attribute *interactionRobot1*. In this case the specialized process interaction *pick and part exchange* is selected for entity node *robot 1* correctly.

3 Automatic Generation of Control Programs

Figure 9 shows the automatic generation of control programs. The starting point is an initial parameter vector P_{init} that contains configuration parameters and relevant process values that are partly mapped to SES attributes. At first the decision component analyzes the control-SES using the parameter vector P and derives a control structure in terms of a parameterized tree, called Pruned Entity Structure (PES).

The derived PES defines a unique control structure, where the leaf nodes present atomic control tasks and process interactions that are stored as parameterizable software components in a model base (MB).

Second the control generation and execution component generates an executable control program considering the information coded in the PES and using components from the MB.

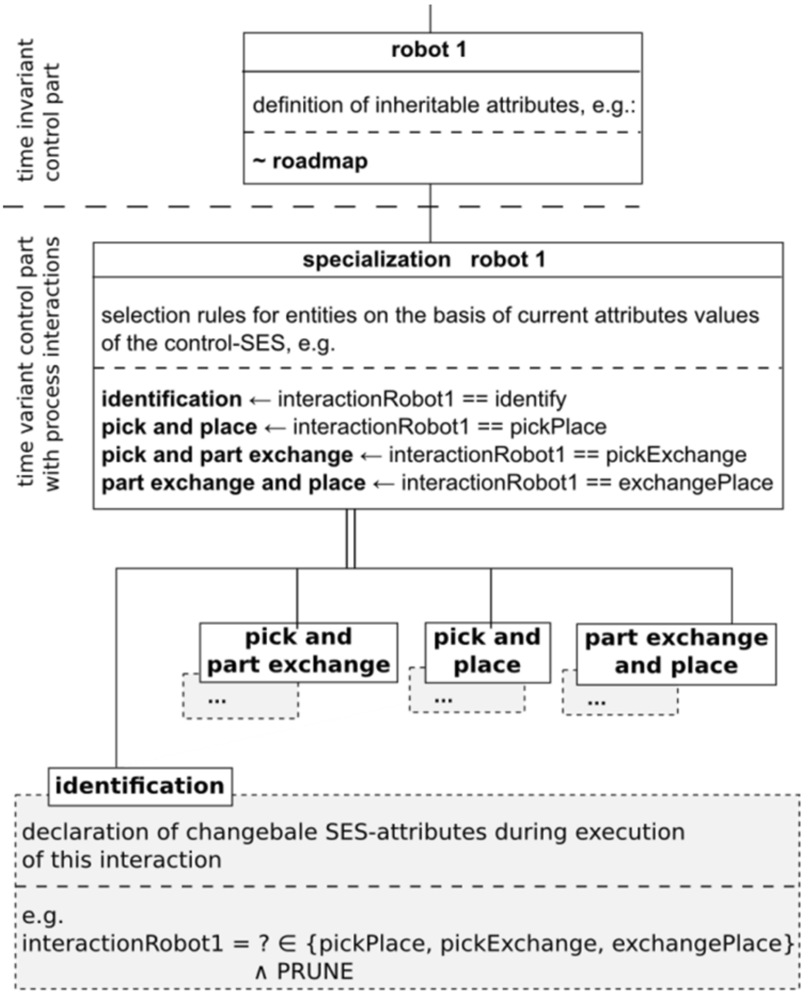


Figure 8. Extended specification of the control-SES for entity node robot 1.

This control program is structured according to the SBC approach in a control model C, a process model P and a process interface I, and is executed by means of a real-time simulator.

During control operation the atomic control tasks and process interactions modify state S of the temporary control according to the real process behavior. A subset of these state changes is monitored by the monitoring component that updates the parameter vector P too.

The decision component maps a subset of P to SES attributes A and decides whether the current temporary control is finished or has to be interrupted. If, so a new control structure is derived by analyzing the SES.

The cycle has to be repeated until the occurrence of any abort criteria. Moreover, Figure 9 illustrates the modularization used by the FTC/SES method. The strict separation of control specification on the one hand and implementation of atomic control tasks and process interactions as parameterizable components on the other supports the development of high flexible controls.

4 Summary

The FTC/SES method introduced has been prototypically implemented and tested in the programming environment MATLAB. The iterative pruning of the control-SES to derive valid control programs is implemented by a MATLAB interface to SWI-Prolog.

The atomic control tasks and process interactions stored in a model base have been implemented in MATLAB based on the DEVS formalism [7, 8]. Hence, each generated control program presents a modular hierarchical DEVS model that is executed by a DEVS simulation environment. This DEVS simulation environment has also been implemented in MATLAB and can be synchronized with real-time.

The application of a cooperating robot control as discussed has been implemented entirely using the introduced FTC/SES method. The interface component of the robot application has been implemented using the MatlabKK-Robotic Toolbox [9].

Finally, we conclude that the FTC/SES method simplifies the service introduction of flexible robot applications because any maintenance of control tasks and process interactions is focused only on strict encapsulated components in the model base.

Next, work will focus on developing further applications using the FTC/SES method to prove the approach.

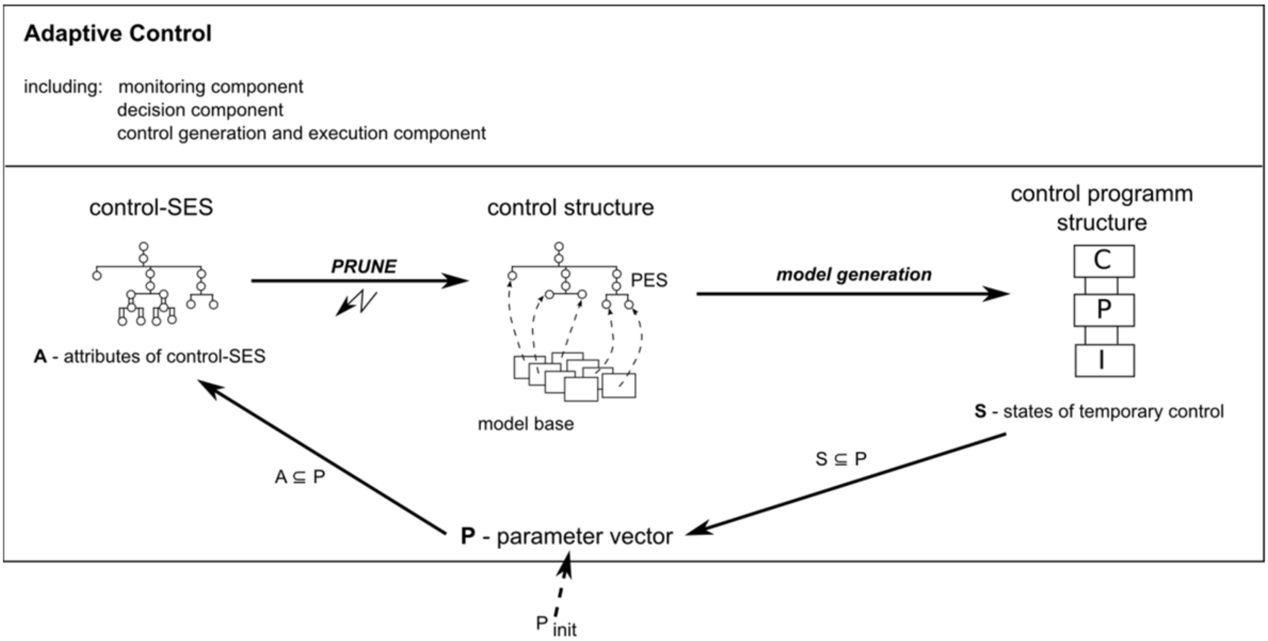


Figure 9. Automatic generation of temporary control programs.

References

[1] F. Breitenecker, D. Solar. *Models, Methods, Experiments - Modern aspects of simulation languages*. In: Proc. 2nd European Simulation Conference, Antwerpen, 1986, SCS, San Diego, 1986, 195 - 199.

[2] W. Jacak. *Intelligent robotic systems: design, planning, and control.* New York: Kluwer Academic / Plenum Publishers, 1998.

[3] M. Haun. *Handbuch Robotik.* Berlin, Heidelberg: Springer Verlag, 2007.

[4] B. P. Zeigler. *Multifaceted Modeling and Discrete Event Simulation.* Academic Press, 1984.

[5] B. P. Zeigler, P. E. Hammonds. *Zodeling and Simulation-based Data Engineering.* Burlington, San Diego, London: Elsevier Academic Press, 2007.

[6] M. Maletzki, T. Pawletta, S. Pawletta, P. Dünow, B. Lampe. *Simulationsmodellbasiertes Rapid Prototyping von komplexen Robotersteuerungen.* atp-Automatisierungstechnische Praxis, Oldenbourg Verlag, München.

[7] D. Abel, A. Bollig. *Rapid Control Prototyping, Methoden und Anwendungen.* Berlin, Heidelberg: Springer Verlag, 2007.

[8] B. P. Zeigler, H. Prähofer, T. G. Kim. *Theory of Modeling and Simulation*.2nd Edition, San Diego, San Francisco, New York, Boston, London, Academic Press 2000.

[9] T. Schwatinski, T. Pawletta, S. Pawletta, C. Kaiser. *Simulation-based development and operation of controls on the basis of the DEVS formalism.* Proceedings of The 7th EUROSIM 2010 Congress, Prag, Czech Republic.

[10] M. Christern, A. Schmidt, T. Schwatinski, T. Pawletta. *KUKA-KAWASAKI-Robotic Toolbox for Matlab.* Hochschule Wismar, http://www.mb.hs-wismar.de/cea/sw_projects.html, 2011.

Submitted: September 2011
Revised: June, 2012
Accepted: August 1, 2012

SNE Simulation News

EUROSIM Data and Quick Info

EUROSIM 2013

8th EUROSIM Congress on Modelling and Simulation

The City Hall, Cardiff, Wales, United Kingdom 10-13 September 2013

www.eurosim2013.info

Contents

Simulation Notes Europe SNE is the official membership journal of EUROSIM and distributed / available to members of the EUROSIM Societies as part of the membership benefits. **SNE** is published in a printed version (Print ISSN 2305-9974) and in an online version (Online ISSN 2306-0271). With **Online SNE** the publisher **ARGESIM** follows the **Open Access** strategy for basic **SNE** contributions. Since 2011 **Online SNE** contributions are identified by DOI 10.11128/sne.xx.nnnnn. for better web availability and indexing.
Print SNE, high-resolution **Online SNE**, and additional SNE contributions are available for subscription via membership in a **EUROSIM** society.

This *EUROSIM Data & Quick Info* compiles data from EUROSIM societies and groups: addresses, weblinks, officers of societies with function and email, to be published regularly in **SNE** issues.

SNE Reports Editorial Board

EUROSIM Khalid Al-Begain, kbegain@glam.ac.uk
Borut Zupančič, *borut.zupancic@fe.uni-lj.si*
Felix Breitenecker, *Felix.Breitenecker@tuwien.ac.at*
ASIM Thorsten Pawletta, *pawel@mb.hs-wismar.de*
CROSSIM Vesna Dušak, *vdusak@foi.hr*
CSSS Mikuláš Alexík, *alexik@frtk.utc.sk*
DBSS A. Heemink, *a.w.heemink@its.tudelft.nl*
FRANCOSIM Karim Djouani, *djouani@u-pec.fr*
HSS András Jávor, *javor@eik.bme.hu*
ISCS M. Savastano, *mario.savastano@unina.it*
PSCS Zenon Sosnowski, *zenon@ii.pb.bialystok.pl*
SIMS Esko Juuso, *esko.juuso@oulu.fi*
SLOSIM Rihard Karba, *rihard.karba@fe.uni-lj.si*
UKSIM Richard Zobel, *r.zobel@ntlworld.com*
CAE-SMSG Emilio Jiminez, *emilio.jiminez@unirioja.es*
LSS Yuri Merkuryev, *merkur@itl.rtu.lv*
ROMSIM Florin Stanciulescu, *sflorin@ici.ro*
RNSS Y. Senichenkov, *sneyb@dcn.infos.ru*
LIOPHANT F. Longo, *f.longo@unical.it*

SNE Editorial Office /ARGESIM

→ *www.sne-journal.org*, *www.eurosim.info*
office@sne-journal.org
Felix Breitenecker, *eic@sne-journal.orgt*
Anna Mathe, *Anna.Mathe@tuwien.ac.at*, *office@sne-journal.org*
Nikolas Popper, *Niki.Popper@drahtwarenhandlung.at*

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

EUROSIM Federation of European Simulation Societies

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

Member Societies. **EUROSIM** members may be national simulation societies and regional or international societies and groups dealing with modelling and simulation. At present **EUROSIM** has thirteen full members and three observer member:

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

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

President	Khalid Al.Begain *kbegain@glam.ac.uk*
Past president	Mikuláš Alexík (CSSS), *alexik@frtk.fri.utc.sk*
Secretary	Borut Zupančič (SLOSIM) *borut.zupancic@fe.uni-lj.si*
Treasurer	Felix Breitenecker (ASIM) *felix.breitenecker@tuwien.ac.at*
SNE Repres.	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*

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

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

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

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

EUROSIM 2013 will be organised by UKSIM in Cardiff, Wales, UK, September 10-13, 2013.

Chairs / Team EUROSIM 2010
Khalid Al.Begain, *kbegain@glam.ac.uk* Richard Zobel, *r.zobel@ntlworld.com* David Al-Dabass, *david.al-dabass@ntu.ac.uk* Alessandra Orsoni, *a.orsoni@kingston.ac.uk* Richard Cant, *richard.cant@ntu.ac.uk*

→ *www.eurosim2013.info*

ASIM
German Simulation Society
Arbeitsgemeinschaft Simulation

ASIM (Arbeitsgemeinschaft Simulation) is the association for simulation in the German speaking area, servicing mainly Germany, Switzerland and Austria. ASIM was founded in 1981 and has now about 700 individual members, and 30 institutional or industrial members. Furthermore, ASIM counts about 300 affiliated members.

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

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

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

ASIM Officers	
President	Felix Breitenecker *felix.breitenecker@tuwien.ac.at*
Vice presidents	Sigrid Wenzel, *s.wenzel@uni-kassel.de*
	T. Pawletta, *pawel@mb.hs-wismar.de*
Secretary	Anna Mathe, *anna.mathe@tuwien.ac.at*
Treasurer	I. Bausch-Gall, *Ingrid@Bausch-Gall.de*
Membership affairs	S. Wenzel, *s.wenzel@uni-kassel.de*
	W. Maurer, *werner.maurer@zhwin.ch*
	I. Bausch-Gall, *Ingrid@Bausch-Gall.de*
	F. Breitenecker, *felix.breitenecker@tuwien.ac.at*
Universities / Research Inst.	S. Wenzel, *s.wenzel@uni-kassel.de*
	W. Wiechert, *W.Wiechert@fz-juelich.de*
	J. Haase, *Joachim.Haase@eas.iis.fraunhofer.de*
	Katharina Nöh, *k.noeh@fz-juelich.de*
Industry	S. Wenzel, *s.wenzel@uni-kassel.de*
	K. Panreck, *Klaus.Panreck@hella.com*
Conferences	Klaus Panreck *Klaus.Panreck@hella.com*
	A. Gnauck, *albrecht.gnauck@tu-cottbus.de*
Publications	Th. Pawletta, *pawel@mb.hs-wismar.de*
	Christina Deatcu, *christina.deatcu@hs-wismar.de*
	F. Breitenecker, *felix.breitenecker@tuwien.ac.at*
Repr. EUROSIM	F. Breitenecker, *felix.breitenecker@tuwien.ac.at*
	N. Popper, *niki.popper@drahtwarenhandlung.at*
Education / Teaching	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
	N. Popper, *niki.popper@drahtwarenhandlung.at*
	Katharina Nöh, *k.noeh@fz-juelich.de*
International Affairs	H. Szczerbicka, *hsz@sim.uni-hannover.de*
	O. Rose, *Oliver.Rose@tu-dresden.de*
Editorial Board SNE	T. Pawletta, *pawel@mb.hs-wismar.de*
	Ch. Deatcu, *christina.deatcu@hs-wismar.de*
Web EUROSIM	Anna Mathe, *anna.mathe@tuwien.ac.at*

Last data update March 2011

ASIM Working Committee. ASIM, part of GI - Gesellschaft für Informatik, is organised in Working Committees, dealing with applications and comprehensive subjects in modelling and simulation:

ASIM Working Committee	
GMMS	Methods in Modelling and Simulation Th. Pawletta, *pawel@mb.hs-wismar.de*
SUG	Simulation in Environmental Systems Wittmann, *wittmann@informatik.uni-hamburg.de*
STS	Simulation of Technical Systems H.T.Mammen, *Heinz-Theo.Mammen@hella.com*
SPL	Simulation in Production and Logistics Sigrid Wenzel, *s.wenzel@uni-kassel.de*
EDU	Simulation in Education/Education in Simulation N. Popper, *niki.popper@drahtwarenhandlung.at*
	Working Groups for Simulation in Business Administration, in Traffic Systems, for Standardisation, for Validation, etc.

CROSSIM – Croatian Society for Simulation Modelling

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

→ *www.eurosim.info*

vdusak@foi.hr

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

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

Last data update March 2011

CSSS – Czech and Slovak Simulation Society

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

→ *www.fit.vutbr.cz/CSSS*

snorek@fel.cvut.cz

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

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

Last data update December 2009

FRANCOSIM – Société Francophone de Simulation

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

- Pole Modelling and simulation of discrete event systems. Pole Contact: *Henri Pierreval, pierreva@imfa.fr*
- Pole Modelling and simulation of continuous systems. Pole Contact: *Yskandar Hamam, y.hamam@esiee.fr*

→ *www.eurosim.info*

y.hamam@esiee.fr

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

FRANCOSIM Officers	
President	Yskandar Hamam, *y.hamam@esiee.fr*
Treasurer	François Rocaries, *f.rocaries@esiee.fr*
Repr. EUROSIM	Yskandar Hamam, *y.hamam@esiee.fr*
Edit. Board SNE	Yskandar Hamam, *y.hamam@esiee.fr*

Last data update April 2006

DBSS – Dutch Benelux Simulation Society

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

→ www.eurosim.info

a.w.heemink@its.tudelft.nl

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

DBSS Officers	
President	A. Heemink, *a.w.heemink@its.tudelft.nl*
Vice president	W. Smit, *smitnet@wxs.nl*
Treasurer	W. Smit, *smitnet@wxs.nl*
Secretary	W. Smit, *smitnet@wxs.nl*
Repr. EUROSIM	A. Heemink, *a.w.heemink@its.tudelft.nl*
Deputy	W. Smit, *smitnet@wxs.nl*
Edit. Board SNE	A. Heemink, *a.w.heemink@its.tudelft.nl*

Last data update April 2006

HSS – Hungarian Simulation Society

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

→ *www.eurosim.info*

javor@eik.bme.hu

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

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

Last data update March 2008

PSCS – Polish Society for Computer Simulation - update

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

→ *www.ptsk.man.bialystok.pl*

leon@ibib.waw.pl

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

PSCS Officers	
President	Leon Bobrowski, *leon@ibib.waw.pl*
Vice president	Andrzej Grzyb, Tadeusz Nowicki
Treasurer	Z. Sosnowski, *zenon@ii.pb.bialystok.pl*
Secretary	Zdzislaw Galkowski, *Zdzislaw.Galkowski@simr.pw.edu.pl*
Repr. EUROSIM	Leon Bobrowski, *leon@ibib.waw.pl*
Deputy	A.Chudzikiewicz, *ach@it.pw.edu.pl*
Edit. Board SNE	Z.Sosnowski, *zenon@ii.pb.bialystok.pl*
PSCS Board Members	R. Bogacz , Z. Strzyzakowski Andrzej Tylikowski

Last data update March 2009

ISCS – Italian Society for Computer Simulation

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

→ *www.eurosim.info*

Mario.savastano@uniina.at

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

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

Last data update April 2005

SIMS – Scandinavian Simulation Society

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

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

→ *www.scansims.org*

esko.juuso@oulu.fi

✉ SIMS / Esko Juuso, Department of Process and Environmental Engineering, 90014 Univ.Oulu, Finland

SIMS Officers	
President	Esko Juuso, *esko.juuso@oulu.fi*
Treasurer	Vadim Engelson, *vadim.engelson@mathcore.com*
Repr. EUROSIM	Esko Juuso, *esko.juuso@oulu.fi*
Edit. Board SNE	Esko Juuso, *esko.juuso@oulu.fi*
Web EUROSIM	Vadim Engelson

Last data update December March 2011

SLOSIM – Slovenian Society for Simulation and Modelling

SLOSIM - Slovenian Society for Simulation and Modelling was established in 1994 and became the full member of **EUROSIM** in 1996. Currently it has 69 members from both slovenian universities, institutes, and industry. It promotes modelling and simulation approaches to problem solving in industrial as well as in academic environments by establishing communication and cooperation among corresponding teams.

→ www.slosim.si

slosim@fe.uni-lj.si

✉ SLOSIM / Rihard Karba, Faculty of Electrical Engineering, University of Ljubljana,
Tržaška 25, 1000 Ljubljana, Slovenia

SLOSIM Officers	
President	B. Zupančič, *borut.zupancic@fe.uni-lj.si*
Vice president	Leon Žlajpah, *leon.zlajpah@ijs.si*
Secretary	Vito Logar, *vito.logar@fe.uni-lj.si*
Treasurer	Milan Simčič, *milan.simcic@fe.uni-lj.si*
Repr. EUROSIM	B. Zupančič, *borut.zupancic@fe.uni-lj.si*
Deputy	Rihard Karba, *rihard.karba@fe.uni-lj.si*
Edit. Board SNE	Rihard Karba, *rihard.karba@fe.uni-lj.si*
Web EUROSIM	Vito Logar, *vito.logar@fe.uni-lj.si*

Last data update March 2011

UKSIM – United Kingdom Simulation Society

UKSIM has more than 100 members throughout the UK from universities and industry. It is active in all areas of simulation and it holds a biennial conference as well as regular meetings and workshops.

→ *www.uksim.org.uk*

david.al-dabass@ntu.ac.uk

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

UKSIM Officers	
President	David Al-Dabass, *david.al-dabass@ntu.ac.uk*
Secretary	A. Orsoni, *A.Orsoni@kingston.ac.uk*
Treasurer	B. Thompson, *barry@bjtcon.ndo.co.uk*
Membership chair	K. Al-Begain, *kbegain@glam.ac.uk*
Univ. liaison chair	R. Cheng, *rchc@maths.soton.ac.uk*
Repr. EUROSIM	Richard Zobel, *r.zobel@ntlworld.com*
Edit. Board SNE	Richard Zobel, *r.zobel@ntlworld.com*

Last data update March 2009 (partially)

CEA-SMSG – Spanish Modelling and Simulation Group

CEA is the Spanish Society on Automation and Control In order to improve the efficiency and to deep into the different fields of automation, the association is divided into thematic groups, one of them is named ‘Modelling and Simulation’, constituting the group.

→ *www.cea-ifac.es/wwwgrupos/simulacion*

→ *simulacion@cea-ifac.es*

CEA-SMSG / María Jesús de la Fuente,
System Engineering and AutomaticControl department,
University of Valladolid,
Real de Burgos s/n., 47011 Valladolid, SPAIN

CAE - SMSG Officers	
President	María J. la Fuente, *maria@autom.uva.es*
Repr. EUROSIM	Emilio Jiminez, *emilio.jiminez@unirioja.es*
Edit. Board SNE	Emilio Jiminez, *emilio.jiminez@unirioja.es*

Last data update March 2009

LSS – Latvian Simulation Society

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

→ *briedis.itl.rtu.lv/imb/*

merkur@itl.rtu.lv

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

LSS Officers	
President	Yuri Merkuryev, *merkur@itl.rtu.lv*
Repr. EUROSIM	Yuri Merkuryev, *merkur@itl.rtu.lv*
Edit. Board SNE	Yuri Merkuryev, *merkur@itl.rtu.lv*

Last data update December 2008

ROMSIM – Romanian Modelling and Simulation Society

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

→ *www.ici.ro/romsim/*

sflorin@ici.ro

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

ROMSIM Officers	
President	Florin Stanciulescu, *sflorin@ici.ro*
Vice president	Florin Hartescu, *flory@ici.ro* Marius Radulescu, *mradulescu@ici.ro*
Secretary	Zoe Radulescu, *radulescu@ici.ro*
Repr. EUROSIM	Florin Stanciulescu, *sflorin@ici.ro*
Deputy	Florin Hartescu, *flory@ici.ro*
Edit. Board SNE	Florin Stanciulescu, *sflorin@ici.ro*

Last data update March 2009

RNSS – Russian Simulation Socitey

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

→ *www.simulation.su*

yusupov@iias.spb.su

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

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, organization of courses and stages in companies to apply the simulation to real problems.

→ *www.liophant.org*

info@liophant.org

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

SNE – Simulation Notes Europe

Simulation Notes Europe publishes peer reviewed *Technical Notes*, *Short Notes* and *Overview Notes* on developments and trends in modelling and simulation in various areas and in application and theory. Furthermore **SNE** documents the **ARGESIM Benchmarks** on *Modelling Approaches and Simulation Implementations* with publication of definitions, solutions and discussions (*Benchmark Notes*). Special *Educational Notes* present the use of modelling and simulation in and for education and for e-learning.

SNE is the official membership journal of **EUROSIM**, the Federation of European Simulation Societies. A News Section in **SNE** provides information for **EUROSIM** Simulation Societies and Simulation Groups.

SNE is published in a printed version (Print ISSN 2305-9974) and in an online version (Online ISSN 2306-0271). With **Online SNE** the publisher **ARGESIM** follows the **Open Access** strategy, allowing download of published contributions for free. Since 2011 **Online SNE** contributions are identified by an DOI (Digital Object Identifier) assigned to the publisher **ARGESIM** (DOI prefix 10.11128). **Print SNE**, high-resolution **Online SNE**, source codes of the *Benchmarks* and other additional sources are available for subscription via membership in a **EUROSIM** society.

Authors Information. 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*):

- *Overview Note* – State-of-the-Art report in a specific area, up to 14 pages, only upon invitation
- *Technical Note* – scientific publication on specific topic in modelling and simulation, 6 – 8 (10) pages
- *Education Note* – modelling and simulation in / for education and e-learning; max. 6 pages
- *Short Note* – recent development on specific topic, max. 4 pages
- *Software Note* – specific implementation with scientific analysis, max 4 pages
- *Benchmark Note* – Solution to an ARGESIM Benchmark;basic solution 2 pages, extended and commented solution 4 pages, comparative solutions on invitation

Interested authors may find further information at **SNE's** website → **www.sne-journal.org** (layout templates for *Notes*, requirements for benchmark solutions, etc.).

ASIM ASIM ASIM

ASIM - Buchreihen / ASIM Book Series

Fortschritte in der Simulationstechnik / Frontiers in Simulation
Monographs - Conference Proceedings

Simulation und Optimierung in Produktion und Logistik.
L. März, W. Krug, O.Rose, G. Weigert , G. (eds.) , ISBN 978-3-642-14535-3, Springer, 2011

Integrationsaspekte der Simulation: Technik, Organisation und Personal.
Zülch, G., Stock, P. (eds.), ISBN 978-3-86644-558-1, KIT Scientific Publishing, Karlsruhe, 2010

Verifikation und Validierung für die Simulation in Produktion und Logistik, Vorgehensmodelle und Techniken.
M. Rabe, S. Spieckermann, S. Wenzel (eds.); ISBN: 978-3-540-35281-5, Springer, Berlin, 2008

Advances in Simulation for Production and Logistics Applications.
M. Rabe (ed.), , ISBN 978-3-8167-7798-4, Fraunhofer IRB-Verlag, Stuttgart, 2008

Modellierung, Regelung und Simulation in Automotive und Prozessautomation - Proceedings 5. ASIM-Workshop Wismar 2011. –
C. Deatcu, P. Dünow, T. Pawletta, S. Pawletta (eds.), ISBN 978-3-901608-36-0, ASIM/ARGESIM, Wien, 2011.

Reihe Fortschrittsberichte Simulation / Series Advances in Simulation

Ch. Steinbrecher: **Ein Beitrag zur prädiktiven Regelung verbrennungsmotorischer Prozesse**
FBS 18, ASIM/ARGESIM Vienna, 2010; ISBN 978-3-901608-68-1, ARGESIM Report 68

O. Hagendorf: **Simulation-based Parameter and Structure Optimisation od Discrete Event Systems**
FBS 17, ASIM/ARGESIM Vienna, 2010; ISBN 978-3-901608-67-4, ARGESIM Report 67

D. Leitner: **Simulation of Arterial Blood Flow with the Lattice Boltzmann Method**
FBS16, ASIM/ARGESIM Vienna, 2009; ISBN 978-3-901608-66-7, ARGESIM Report 16

Th. Löscher: **Optimisation of Scheduling Problems Based on Timed Petri Nets.**
FBS 15, ASIM/ARGESIM Vienna, 2009; ISBN 978-3-901608-65-0, ARGESIM Report 15

J. Wöckl: **Hybrider Modellbildungszugang für biologische Abwasserreinigungsprozesse.**
FBS 14, ASIM/ARGESIM Vienna, ISBN 3-901608-64-8, 2006, , ARGESIM Report 14,

M. Gyimesi: **Simulation Service Providing als Webservice zur Simulation Diskreter Prozesse.**
FBS 13, ASIM/ARGESIM Vienna, ISBN 3-901-608-63-X, 2006, ARGESIM Report 13

R. Fink: **Untersuchungen zur Parallelverarbeitung mit wissenschaftlich-technischen Berechnungsumgebungen.**
FBS 12, ASIM/ARGESIM Vienna, 2008; ISBN 978-3-901608-62-9, ARGESIM Report 12

H. Ecker: **Suppression of Self-excited Vibrations in Mechanical Systems by Parametric Stiffness Excitation.**
FBS 11, ASIM/ARGESIM Vienna, ISBN 3-901-608-61-3, 2006, ARGESIM Report 11

Orders:
ASIM/ARGESIM Office Germany, Hochschule Wismar, PF 1210, 23952 Wismar, Germany
ASIM/ARGESIM Geschäftsstelle Österreich, c/o DWH, Neustiftgasse 57, 1040 Vienna, Austria

Order and Download via ASIM webpage in preparation

Info: www.asim-gi.org, info@asim-gi.org

REPORTS

REPORTS

EUROSIM 2013

8th EUROSIM Congress on Modelling and Simulation

The City Hall, Cardiff, Wales, United Kingdom 10-13 September 2013

EUROSIM Congresses are the most important modelling and simulation events in Europe. For EUROSIM2013, we are soliciting original submissions describing novel research and developments in the following (and related) areas of interest: Continuous, discrete (event) and hybrid modelling, simulation, identification and optimization approaches. Two basic contribution motivations are expected: M&S Methods and Technologies and M&S Applications. Contributions from both technical and non-technical areas are welcome.

Congress Topics

The EUROSIM 2013 Congress will include invited talks, parallel, special and the poster sessions. The Congress topics of interest include, but are not limited to:

Intelligent Systems and Applications
Hybrid and Soft Computing
Communication Systems and Networks
Case Studies, Emergent Technologies
Workflow Modelling and Simulation
Web-based Simulation
Security Modelling and Simulation
Computer Games and Simulation
Neural Networks, Fuzzy Systems & Evolutionary Computation
Autonomous Mental Development
Bioinformatics and Bioengineering
Circuits, Sensors and Devices

e-Science and e-Systems
Image, Speech & Signal Processing
Human Factors and Social Issues
Industry, Business, Management
Virtual Reality, Visualization and Computer Games
Internet Modelling, Semantic Web and Ontologies
Computational Finance & Economics
Systems Intelligence and Intelligence Systems
Adaptive Dynamic Programming and Reinforcement Learning

Methodologies, Tools and Operations Research
Discrete Event /RT Systems
Mobile/Ad hoc wireless networks, mobicast, sensor placement, target tracking
Control of Intelligent Systems and Control Intelligence
Robotics, Cybernetics, Control Engineering, & Manufacturing
Energy, Power, Transport, Logistics, Harbour, Shipping and Marine Simulation
Semantic & Data Mining

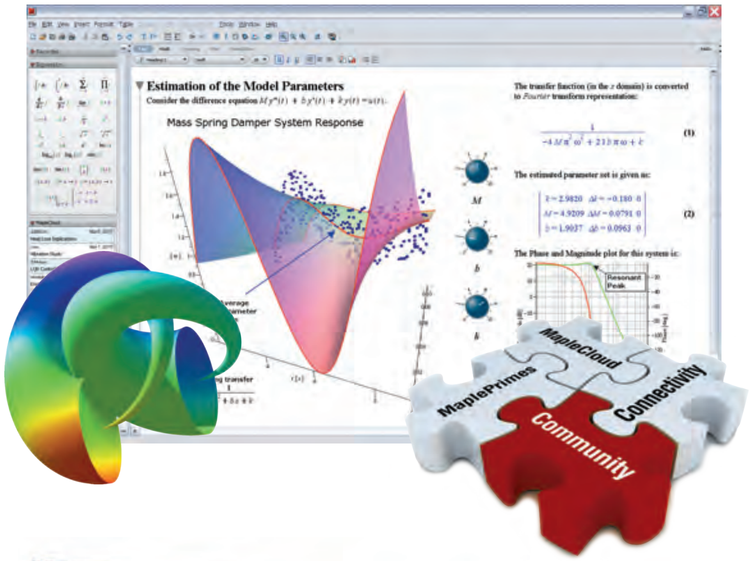
Congress Venue / Social Events

The Congress will be held in the historic and magnificent City Hall in the heart of Cardiff, the capital city of Wales. The Gala Dinner will be held in the main hall of the National Museum of Wales. Social activities include visits to Cardiff Castle and Caerphilly Castle.

Congress Team: K. Al-Begain, A. Orsoni, R. Zobel, R. Cant, D. Al-Dabass; *kbegain@glam.ac.uk*

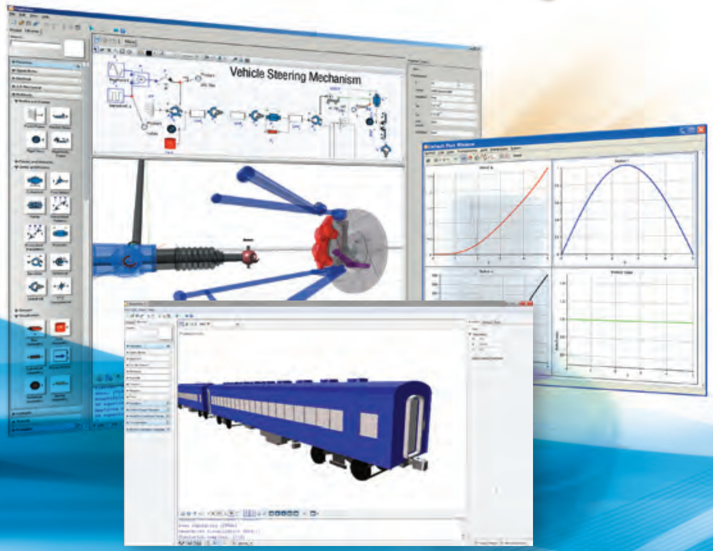
Info: www.eurosim2013.info

New challenges in modeling and simulation require powerful tools



Maple™ 15

The result of over 30 years of cutting-edge research and development, Maple combines the world's most powerful mathematical computation engine with an intuitive, "clickable" user interface.



MapleSim™ 5

MapleSim is a physical modeling and simulation tool built on a foundation of symbolic computation technology.

Get a Price Quote:
www.maplesoft.com/sne

Contact us:

+49 (0)241/980919-30

Maplesoft
Mathematics • Modeling • Simulation
A Cybernet Group Company

www.maplesoft.com | germany@maplesoft.com