View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by computer science publication server

Agent-based modeling and simulation of individual traffic as an environment for bus schedule simulation

Daniel Lückerath
Department of Computer Science, University of Cologne, Albertus-Magnus-Platz, D-50923, Cologne, Germany

Oliver Ullrich, Ingo Proff, Patrick Kuckertz, Ewald Speckenmeyer
Department of Computer Science, University of Cologne, Albertus-Magnus-Platz, D-50923, Cologne, Germany

Abstract

To re-establish the regular driving operations of a tram network, which was disturbed significantly by unforeseen external events, traffic schedulers apply rescheduling and rerouting strategies. These strategies are usually multi-modal; they consider the interaction of trams, buses, even taxis. Thus, to evaluate the applicability of a given rescheduling or rerouting strategy prior to its implementation in the real-world system, a multi-modal simulation software is needed. In this article we present an agent-based model of individual traffic which will be applied as background to a planned simulation of bus traffic. These combined models are to be integrated with an existing tram schedule simulation; the resulting multi-modal model will then be applied to evaluate the usefulness of given rescheduling or rerouting strategies. After a short introduction to agent-based modeling and simulation, as well as to existing models of individual traffic, this paper proposes to model the behavior of individual traffic as an environment for agent-based bus schedule simulation. Finally, some experiments are conducted by modeling and simulating individual traffic in Cologne's highly frequented Barbarossaplatz area.

1 Introduction

In many tram networks multiple lines share tracks and stations, thus requiring robust schedules which prevent inevitable small delays from spreading through the network. Feasible schedules also have to fulfill various planning requirements originating from political and economic reasons. In [8] and [17], some of the authors present a tool set designed to generate tram schedules optimized for robustness, which also satisfy given sets of planning requirements. These tools are aimed to assist traffic schedulers to compare time tables with respect to their applicability and evaluate them prior to their implementation in the field.
We now want to extend the tool set to consider larger disturbances, e.g. originating from broken down trams, closed stations, or other blocked resources. Careful schedule design is not sufficient to handle these major disturbances, traffic operators have to apply rescheduling and rerouting strategies (see [7] and [9]) to reestablish regular operations. To be effective, these strategies are inevitably multi-modal: trams are rescheduled to compensate for cancellations, buses are rerouted to relieve the tram network, some traffic operators even co-operate with taxi companies (see [18]). To evaluate the applicability of a given rescheduling or rerouting strategy prior to its implementation in the real-world system, a multi-modal simulation software is needed.
This paper presents a first step towards building that simulation engine. To map multi-modal traffic, a model of bus traffic will be incorporated into the existing tram simulation. Buses are highly dependent on surrounding individual traffic, so a valid model has to include at least a coarse representation of its behavior. Because we aim at an agent-based model of multi-modal traffic, individual traffic can be seen as the environment in which buses - modeled as agents - act and interact according to their set of rules and strategies. Central goal of this

paper is to present the design and implementation of an agent-based model of individual traffic, to be used as an environmental layer to a bus traffic simulation yet to be developed.

The remainder of this paper begins with a presentation of some background on agent-based modeling and simulation and the simulation of individual traffic (section 2). We continue with presenting a model of individual traffic as an environment for agent-based bus schedule simulation (section 3). Based on an implementation of this model, some experiments are conducted, focusing on Cologne's Barbarossaplatz area, a highly frequented part of Cologne's inner city street network (section 4). The paper closes with a short summary of the lessons learned and some thoughts on future work (section 5).

2 Background

2.1 Agent-based modeling and simulation

Agent-based modeling and simulation (ABMS) is a comparatively new approach of modeling complex systems as sets of autonomous, interacting agents (for this section see [10]). The behavior of the agents is determined by state variables (attributes) and sets of internal rules. The individual agents act on local information and interact with a subset of other agents and the environment they exist in. In many cases, self-organization can be observed, patterns and structures emerge that were not explicitly modeled. The approach has a broad range of applications in various fields of research, including the study of pastoral-nomadic land use systems (see [6]), of the human immune system (see [4]), of the population growth and collapse of ancient civilizations (see [1]), and of public and individual traffic systems (e.g. see [3], [14], and [15]).
An agent-based model usually includes three components: The agents, their interaction rules, and the agents' environment.
The agents are usually self-contained and autonomous; they have attributes whose values change over the course of the simulation run. Their behavior is determined by a set of rules, and they interact dynamically with other agents and the environment they exist in. In more complex models, agents are often goal-directed and adaptive, and may even be heterogeneous.
Because agents interact only with a local subset of other agents, their neighbors, only local information is available to them. The agents that are part of a neighborhood are usually determined by the model's topology, which e.g. might be a (static or dynamic) network, a spatial grid, or an aspatial “soup” model. The members of an agent's neighborhood may change rapidly during the simulation run.
In addition to their communication with their neighbors, agents also interact with their environment. This interaction may only provide basic information, like the position of the agents in a spatial model (e.g. street lanes and crossings in a traffic model). It may also provide detailed information, like the capacity of and the maximum velocity on lanes, or the state of embedded street lights. An agent's environment is often built as a complex simulation model itself, e.g. based on cellular automata (again see [6]).

2.2 Simulation of individual traffic

Several approaches to model individual traffic are known, many of them based on cellular automata (e.g. see [11], [12], and [13]), or based on ABMS (e.g. see [3], [14], and [15]).
In [12], Nagel and Schreckenberg present a model for freeway traffic, based on simple cellular automata. The model utilizes a set of very simple rules: If there is a free lane ahead, each car c_i tries to accelerate up to a certain maximum velocity. If another car c_j is registered ahead, c_i decreases its velocity. For randomization, the car's velocity is decreased with a

small probability. Lane-switching and overtaking are not possible in this model; it only maps one single lane. Even with its very simple rule set, the model shows some non-trivial and realistic behavior: Results show that up from a certain traffic density, traffic jams develop without an external cause, moving backwards through the model, very much alike observed real-world behavior. In [13] Nagel, et al. extend this approach to multi-lane traffic.
Based on the Nagel/Schreckenberg approach, Moltenbrey and Bungartz (see [11]) aim at simulating real world situations. Their model includes lane-switching and heterogeneous vehicles, many of them filling more than one node of the cellular grid. Bicycles and motorcycles are included with an unique behavior and are not just modeled as slower cars. Each vehicle follows its individual activity plan: It has a pre-planned route from trip origin to destination, calculated with a shortest-path algorithm. This algorithm considers waiting periods resulting from traffic jams, and dynamically chooses alternative routes. The model also includes public transportation, modeling time tables as special activity plans. Because of all these points, the model is necessarily very complex, and stretches the paradigm of cellular automata.

Several agent-based models (of different complexity) of individual traffic are known, each fitted for its special application.
Ehlert and Rothkrantz propose a multi-agent model (see [3]) of urban individual traffic. Their agents are quite complex and capable of what they call “tactical-level driving”. The agents are designed modular; each module takes part in the decision process and can be adapted or replaced. Those modules include a sensor module, memory for storing data, a controller for regulating access to the memory, a short-term planner, multiple sets of behavior rules, and an arbiter which selects the best action proposed by the behavior rules.
Paruchuri, Pullalarevu and Karlapalem (see [14]) present a model targeted at simulating individual traffic in Indian metropolises, which they term as “chaotic”. Therefore no traffic lights or global overtaking rules are modeled. They try to map a realistic behavior of different driver types, and therefore include several psychological traits. The driver types have attributes like favorite speed, preferred values of acceleration and deceleration, and individual reaction time. To further enhance the model's behavior, the simulation engine includes a relatively complex model of the involved physics. Among other things, the results show a positive correlation between number of aggressive drivers and average speed.
Seele, et al. (see [15]) describe an agent-based simulation of individual traffic as an environment for a human-in-the-loop bicycle simulation. They are therefore interested in a sufficiently realistic behavior. The agents are required to comply to traffic rules, but should also be able to act irrationally and break those rules, so that human participants can experience a sense of danger. Therefore cognitive processes are modeled, based on psychological personality profiles. This approach yields complex agents which take up a lot of computing power. The agents act and interact in a real time environment, which sets a hard upper limit to acceptable processor time. On the other hand, this allows for a complex model, because the simulator does not aim at an as-fast-as-possible speed, or analytical results obtained by a high number of simulation runs.

3 Modeling individual traffic

Because our model of individual traffic will be applied as a backdrop for a model of time table based bus traffic, and thus cannot use up much processor time, we aim for simplicity. From the perspective of the bus agents to be embedded, the individual traffic's swarm behavior has to be represented accordingly. This means that the behavior of each single agent does not necessary have to be mapped in detail, decision processes can therefore be simplified. For these reasons, the proposed method will be less complex than the techniques shown in section 2.2.

3.1 Modeling lanes and crossings

The street network is modeled as an attributed graph, with directed edges representing lanes, and nodes representing splits, joins, traffic lights and other focal points.
Edges have attributes like length, and up to two neighboring lanes (left and right), which go in parallel. Switching lanes is allowed between direct neighbors, as long as the agents find enough free space on the neighboring lane.
As proscribed by the topology of the observed system, nodes dissect lanes into parts of maximal length. If a lane is dissected by a node, all neighboring lanes also have to be dissected to keep the transitive closure intact.
There are three types of nodes: Intersections, joins, and splits. Intersections are nodes with up to one incoming and up to one outgoing edge, and thus without the opportunity to change direction. Some of these are applied as entry points or exit points of the model. Joins are nodes with more than one incoming lane and one outgoing lane. Splits are nodes with one or more incoming lanes and more than one outgoing lane. Here, it is possible for the vehicle to choose one of the outgoing lanes, based on a given set of probabilities.
Nodes can be blocked when vehicles on incoming lanes have a precedence in the right of way. Some nodes include traffic lights, blocking and freeing access to their outgoing edges. This is accomplished by a basic event-based system (as described in [2]), which administrates the light switching events. Lights switch between phases of red and green, following observed time intervals. The lowest common multiple of these phases is called cycle time t_c. The described event mechanism is also utilized to schedule the generation of new vehicles at certain entry points.

3.2 Modeling vehicles

The neighborhood of each agent is described in a predecessor graph, which maps the relationship between each vehicle and its preceding and succeeding car. Because the predecessor/successor relationship also has to work for two vehicles on different lanes, this graph is not always symmetric. The relationship has to be updated each time a vehicle passes onto another edge, either by moving over a node or by switching the lane. Thus, each agent knows which vehicle constitutes its immediate predecessor, and can therefore adapt its speed according to the current distance and speed of that car.
A vehicle is represented by an agent which includes a set of attributes like speed, position, size, and preferred acceleration and deceleration. The agent's rule set can be discerned in three areas: Accelerating and braking, changing direction, and switching lanes.

Accelerating and braking: For simplicity we assume uniform acceleration for all vehicles. At each step of simulation time, the agent has the opportunity to change its velocity, depending on the current distance d_0 to the agent's predecessor or an upcoming traffic light. If d_0 is greater than a lookahead l_{max}, the obstacle is ignored and the agent accelerates with its preferred value, up to its maximum velocity. If d_0 is less than or equal to the lookahead, and if the obstacle moves faster than the agent, an acceleration value is calculated so the minimum security distance is not violated. If the obstacle has a slower speed than the agent, a deceleration value is calculated so that the minimum distance S_D can be reached and the agent has the same velocity as its predecessor at that point. With v_0 as the agent's velocity, w_0 as the obstacle's velocity, and *b* as the obstacle's acceleration, the agent's acceleration *a* can be computed as:

$$a = \frac{1}{2} * \frac{(w_0 - v_0)^2}{S_D - d_0} + b$$

To achieve a higher degree of realism, the agent will only initiate a deceleration if *a* is lesser than a deceleration threshold $r_d < 0$. As an example it could be assumed that a driver going at 4 km/h who notices an obstacle 70 meters away, would not immediately brake to reach a velocity of zero at the obstacle. Instead she would probably accelerate for a while and then brake sharper, thus reaching the other car's position in a shorter time.

Changing direction: If an agent gets to a split node it has to choose one of the outgoing edges. To accomplish this in a realistic way, each dissection features a probability table; its values are derived by observing the real-world crossing. There is a small probability of an agent to move in a circle and thus to never leave the model in the course of the simulation run. Though this behavior would not be realistic (or would it?), for our purposes this is negligible: A few non-realistic vehicles would not compromise the resulting swarm behavior.

Switching lanes: An agent tests whether switching a lane would be safe and advantageous. Thus, two conditions, the security condition and the gain condition have to be fulfilled. *Security condition:* An agent can only switch lanes if there is sufficient free space on the target lane. Additionally, the agent can only change its speed up to a given amount v_{change} to match its velocity to the hypothetical successor and predecessor on the target lane. *Gain condition:* A lane-switch is seen as an advantage, if 1) the possible acceleration a' on the target lane would be greater than the possible acceleration a on the current lane, and 2) the gain of $(a' – a) > 0$ is not counterbalanced by a loss $(b' – b) < 0$ of the successor agent on the target lane. This comparison is balanced by the agent's individual "altruistic factor" τ and a minimal gain parameter $\lambda > 0$, which prevents lane-switching for a minimal gain. Thus, lane-switching is supposed to be advantageous only, if $(a' – a) + \tau^*(b' – b) > \lambda$.

4 Experiments

4.1 Modeling Cologne's Barbarossaplatz area

The Barbarossplatz area (see figure 1) is a highly frequented part of Cologne's inner city network. Here the major arteries of Hohenstauffenring/Salierring, Luxemburger Straße/Weyerstraße, and Roonstraße meet; they are joined by minor roads like Mauritiuswall and Kyffhäuserstraße. Barbarossaplatz is also one of the hubs of Cologne's public transport network, so that buses and trams cross the area periodically.
The area was modeled with 88 nodes and 58 lanes of an accumulated length of 6,979 meters (see figure 2 and table 1). There are eight traffic lights, with a traffic light cycle time t_c of 105 seconds. The vehicle numbers at the entry points were measured at a typical workday evening (see table 2), tables of probabilities for direction changing were also derived from observations.
As major simulation parameters we set a maximum velocity v_{max} of 54 kilometers per hour, a maximum speed change for lane-switching v_{change} of 18 kilometers per hour, a maximum lookahead for obstacles l_{max} of 70 meters, and a driver's response time of 0.5 seconds. The application is then run to simulate a time interval of 240 minutes.

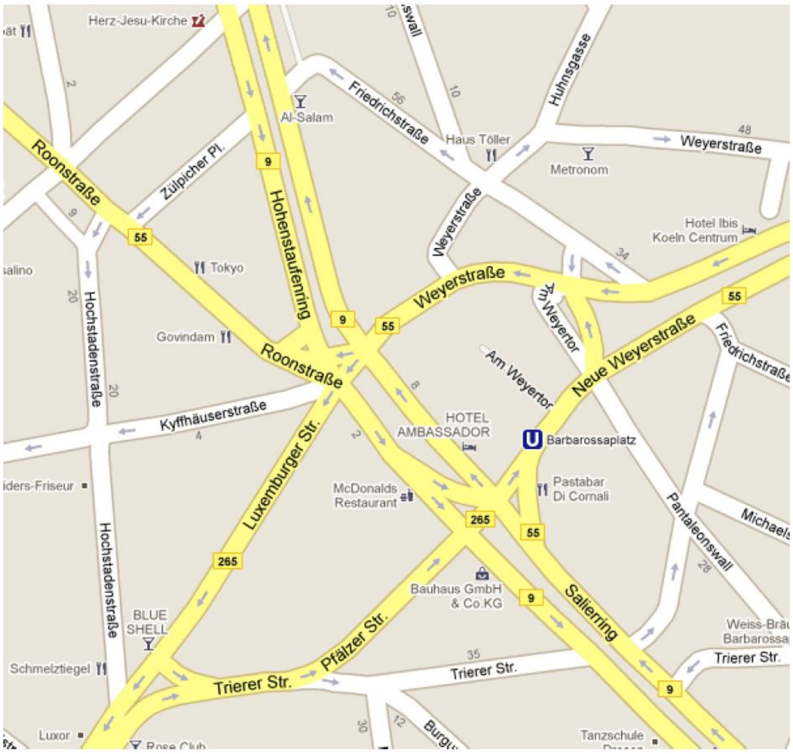

Figure 1: Cologne's Barbarossaplatz area (Source: [5])

4.2 Results and discussion

In the course of a typical simulation run, 18,623 vehicles were generated, they employed an average speed of 19.1 kilometers per hour.

The model's overall behavior seems realistic: The numbers of agents at the entry points are direct results of the simulation parameters and show therefore almost no variations (see table 2). The numbers of leaving agents at the exit points (see table 3) result from the path through the model chosen by individual agents and therefore show some variations. These can be partially explained by the mode of the observation: The numbers of cars at the entry and exit points were gathered sequentially, not simultaneously at all roads.

The simplicity of the model yields some constraints: An agent's path through the model is composed by sequential, independent, and randomized picks without any overriding strategy. Thus, though single decisions are modeled after observations, and therefore match reality, some agents' long-term behavior does not. While the lane-switching behavior seems plausible, it also clearly shows a missing strategy. Agents can be observed to switch lanes in front of traffic lights even if the queue at their current lane is shorter than the one at the target lane, if the predecessor on the target lane is still moving and the one at the current lane is not.

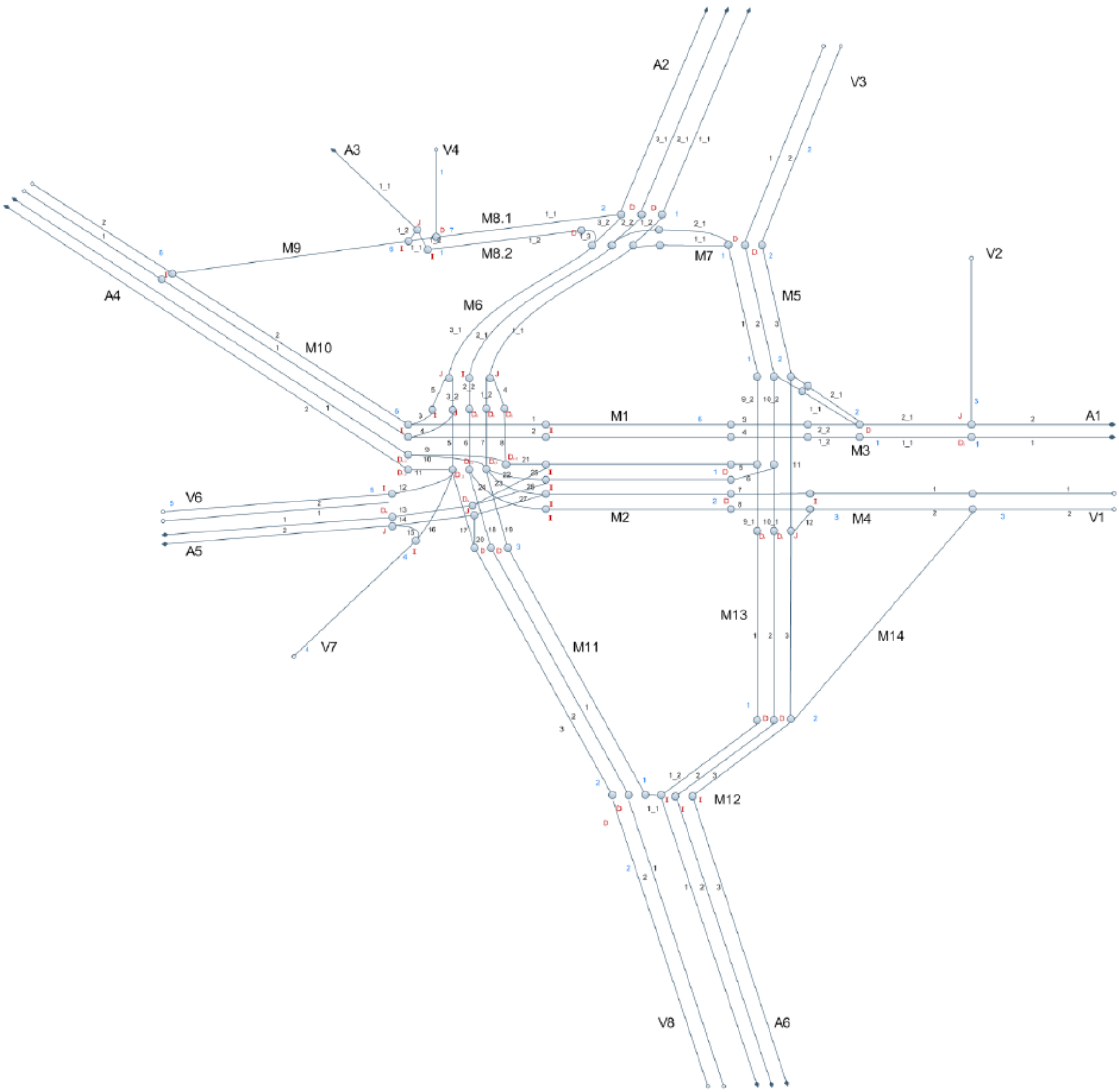

Figure 2: Traffic lane model of the Barbarossaplatz area

No.	N/O Lanes	Lengths
A1	2	12
A2	3	230
A3	1	120
A4	2	170
A5	2	160
A6	3	200
V1	2	150
V2	1	100
V3	2	150
V4	1	60
V5	2	50
V6	2	160
V7	1	160
V8	2	200

No.	N/O Lanes	Lengths
M1	2	100
M2	4	110
M3	2	90
M4	2	90
M5	3	70
M6	3	120
M7	2	55
M8	1	105
M9	1	110
M10	2	120
M11	3	140
M12	3	55
M13	3	95
M14	1	140

Table 1: Number and lengths of lanes

Entry points		Cars per second	
No.	**Name**	**Observed**	**Simulated**
A1	Salierring	0.18	0.16
A2	Neue Weyerstraße	0.50	0.48
A3	Weyerstraße	0.04	0.05
A4	Hohenstaufenring	0.23	0.20
A5	Roonstraße	0.18	0.19
A6	Luxemburger Straße	0.19	0.17
Total		1.32	1.29

Table 2: Numbers of observed and simulated cars at entry points

Exit points		Cars per second	
No.	**Name**	**Observed**	**Simulated**
V1	Salierring	0.15	0.19
V2	Pantaleonsmühlengasse	0.00	0.00
V3	Neue Weyerstraße	0.40	0.40
V4	Mauritiuswall	0.01	0.01
V5	Hohenstaufenring	0.15	0.14
V6	Roonstraße	0.18	0.11
V7	Kyffhäuserstraße	0.02	0.03
V8	Luxemburger Straße	0.36	0.39
Total		1.27	1.28

Table 3: Numbers of observed and simulated cars at exit points

5 Summary and further research

This article presents an approach to model and simulate individual traffic as an environmental layer for the simulation of time table based public bus systems. Following an introduction to the goals and context of our work, it presents some background of ABMS in general and the modeling of individual traffic in particular. We then demonstrate our modeling approach and apply it to Cologne's Barbarossaplatz area.
As a further step of our research, we will develop an agent-based representation of time table based bus traffic which utilizes the proposed model as a background. The combined models will then be embedded with the already existing model of tram traffic (described in [16]) into a common simulation application. This multi-modal application will then be utilized to represent our hometown Cologne's public transportation system.

References

[1] Axtell, R. L., Epstein, J. M., Dean, J. S., Gumerman, G. J., Swedlund, A. C., Harburger, J., Chakravarty, S., Hammond, R., Parker, J., Parker, M.: Population growth and collapse in a multiagent model of the Kayenta Anasazi in Long House Valley. In: Proceedings of the National Academy of Sciences (PNAS), Volume 99, Suppl. 3, 2002, pp. 7275–7279.

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

[3] Ehlert, P. A. M., Rothkrantz, L. J. M.: Microscopic traffic simulation with reactive driving agents. In: Proceedings of 2001 IEEE Intelligent Transportation Systems Conference, Oakland (CA), USA, 2001, pp. 861-866.

[4] Folcik, V. A., An, G. C., Orosz, C. G.: The Basic Immune Simulator: An agent-based model to study the interactions between innate and adaptive immunity. In: Theoretical Biology and Medical Modelling, Volume 4, Number 39, 2007, http://www.tbiomed.com/content/4/1/39, accessed on 27th of April, 2013.

[5] Google Inc: Google Maps, http://maps.google.de. Retrieved on August, 15th, 2011.

[6] Kuckertz, P., Ullrich, O., Linstädter, A., Speckenmeyer, E.: Agent based modeling and simulation of a pastoral-nomadic land use system. In: Simulation Notes Europe (SNE), Volume 21, Number 3-4, Dezember 2011, pp. 147-152.

[7] Lückerath, D.: Thoughts on restauration of regular tram operation. In: Proceedings of Sommertreffen Verkehrssimulation 2012, AM 143, ARGESIM/ASIM Pub., TU Vienna, 2012, pp. 4-6.

[8] Lückerath, D., Ullrich, O., Speckenmeyer, E: Modeling time table based tram traffic. In: Simulation Notes Europe (SNE), ARGESIM/ASIM Pub., TU Vienna, Volume 22, Number 2, August 2012, pp. 61-68.

[9] Lückerath, D., Ullrich, O., Speckenmeyer, E.: Applicability of rescheduling strategies in tram networks. In: Proceedings of ASIM-Workshop STS/GMMS 2013 - ARGESIM Report 41, ASIM-Mitteilung 145, ARGESIM/ASIM Pub., TU Vienna, 2013, pg. 7.

[10] Macal, C. M.; North, M. J.: Tutorial on agent-based modelling and simulation. In: Journal of Simulation 4 (2010), pp. 151-162.

[11] Moltenbrey, M., Bungartz, H.-J.: Design and Implementation of a Fine Grain Microscopic Traffic Simulator with Integrated Timetable-Based Public Transportation. In: Proceedings of ASIM 2006, ARGESIM/ASIM Pub., TU Vienna, 2006.

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

[13] Nagel, K., Wolf, D. E., Wagner, P., Simon, P.: Two-lane traffic rules for cellular automata - A systematic approach. In: Physical Review E, Volume 28, Issue 2, August 1998, pp. 1426-1437.

[14] Paruchuri, P., Pullalarevu, A. R., Karlapalem, K.: Multi Agent Simulation of Unorganized Traffic. In: Proc. AAMAS 2002, Bologna, Italy, 2002, pp. 176-183.

[15] Seele, S., Herpers, R., Bauckhage, C., Becker, P.: Cognitive Agents with Psychological Personality for Traffic Simulations in Virtual Environments. In: Proceedings of Sommertreffen Verkehrssimulation 2012, AM 143, ARGESIM/ASIM Pub., TU Vienna, 2012, pp. 8-12.

[16] Ullrich, O.: Modellbasierte Parallelisierung von Anwendungen zur Verkehrssimulation - Ein dynamischer und adaptiver Ansatz. Dissertation, Univ. Köln, to appear.

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

[18] Zeng, A.Z., Durach, C.F., Fang, Y.: Collaboration decisions on disruption recovery serivce in urban public tram systems. In: Transportation Research Part E, 48, 2012, pp. 578-590.