Chapter 7
Traffic Simulation with SUMO – Simulation of Urban Mobility

Daniel Krajzewicz

7.1 Introduction

"*S*imulation of *U*rban *MO*bility" (Krajzewicz et al., 2002; Krajzewicz et al., 2004; SUMO, 2001–2009), or "SUMO" for short, is a microscopic road traffic simulation. The work on SUMO's design began in the year 2000, with the first implementation being started in the year 2001. In the beginning, SUMO was developed in collaboration between the Center for Applied Informatics Cologne (ZAIK) and the Institute of Transportation Systems (ITS), at the German Aerospace Center (DLR). Since 2004, the work on SUMO is continued at the DLR only, though with contribution from external organizations or individuals.

SUMO is available as "open source" under the GNU General Public License (GPL, 2009), both as source code and in compiled, executable form for multiple Windows and Linux platforms. The reason for building an open-source traffic simulation was twofold. While working on traffic simulations within the academic field, it was noted that many different, small simulations were developed as tools within diploma or doctoral theses, in order to evaluate the objective that was the thesis' real topic. Often, these simulations were incomplete due to the large amount of problems that must be solved for having a complete traffic simulation, and after the thesis was completed, the used simulations were not made available to the public – they disappeared. So on the one hand, the work to be done to have a proper traffic scenario being simulated was repeatedly redone, wasting resources and distracting from the actual topic of interest. On the other hand, the results of such scientific work are hardly comparable as long as the simulations used to evaluate them have different features, use different (sub-) models and implementation details. Because one of the main research areas of the Institute of Transportation Systems was to compare and evaluate different models and algorithms related to traffic and traffic management, the idea was created to develop a free, extensible traffic simulation, so that

D. Krajzewicz (✉)
Institut für Verkehrssystemtechnik, Deutsches Zentrum für Luft- und Raumfahrt e.V., Rutherfordstr. 2, 12489 Berlin, Germany
e-mail: daniel.krajzewicz@dlr.de

J. Barceló (ed.), *Fundamentals of Traffic Simulation*,
International Series in Operations Research & Management Science 145,
DOI 10.1007/978-1-4419-6142-6_7,

(a) different people may use it as a base test bed for their own applications and (b) these applications would get more comparable.

This idea and the academic context SUMO is developed within highly influenced design criteria and SUMO's features. As main design criteria, the following were stated:

- Portability
  The simulation must run on any common environment, because many research organizations were using Linux or – at the time the work on SUMO started – Solaris as operating system.
- Extensibility
  The simulation must be open and easy to understand, so that someone who has not contributed to the original development can adapt it fast to his or her own needs.

In addition, the first application SUMO was meant to be used for was the simulation of traffic based on the daily activities of a synthetic population of the city of Cologne. This synthetic population – together with their activities – was based upon daily activity reports collected during the "Zeitbudgeterhebung 1992" project (Blanke et al., 1996). Each modeled person has certain mobility wishes and adapts his/her behavior regarding these and the possibilities to accomplish them. A common example is a person who has to re-plan his/her afternoon activities due to a jam which made him/her come too late to work. The process of generating this population is described in Hertkorn and Kracht (2002), Hertkorn and Wagner (2004), and Hertkorn (2004). For applying SUMO within this research, additional features of the software were needed:

- Small Memory Footprint
  The simulation must be able to handle scenarios covering large city areas on a standard work station.
- Execution Speed
  Simulation of large areas must be performed as fast as possible.

These quality requests strongly influenced and still influence SUMO's design and implementation, but they are only the first set of conditions that shape SUMO's development. The second is that SUMO has been developed continuously for eight years now, but only in the frame of the projects the Institute of Transportation Systems or external contributors are working at. During this time, SUMO had been improved significantly, but only along the features needed by the actually done projects. For short, a road map for building the best traffic simulation ever does not exist. As a result, the simulation was "incomplete" during the first years (and still is), simply because some features one would expect were not needed for the current research.

The third influence is the fact that most of the users are not traffic scientists but computer scientists. There may be several reasons for this. At first, SUMO is not an established or even certified traffic simulation as others may be. Furthermore, a user must have a high affinity toward computer systems, meaning that she/he should be able to work on the command line with no graphical support, and, in some cases, even write own programs for preparing the inputs and evaluating the outputs. Also, we assume that traffic scientists, consultants working in this area, or traffic administrations already possess a traffic simulation and/or are willing to spend money on an established system. This is often not the case if a single diploma thesis in computer science shall be written. The lack of users with a traffic science background has a strong effect on the feedback we get. Only few questions arise about the used models or are in any means related to research on traffic simulations as such.

Nonetheless, the aimed quality criteria could be achieved and we were able to use SUMO successfully in several projects over the past years. The addressed questions range over a set of traffic management aspects, from evaluating new adaptive traffic lights for single intersection control to monitoring and forecasting traffic within large-scale areas. Possible usages will be briefly described by examples in the following.

Being a major use case for traffic simulations, optimization of traffic lights was also addressed within two of our projects, though with two different granularities. Within the project "OIS" – "optical information systems" – a new control algorithm for an intersection was developed and evaluated (Krajzewicz D et al., 2005). This algorithm uses information gained by recognizing vehicles within images taken via cameras located at the controlled intersection. Using this recognition, the system is capable to compute the queue lengths in front of the traffic lights and the implemented algorithm uses these lengths to decide which stream should get green for a longer time. The simulation's task was to evaluate this algorithm's performance in comparison to the real world's traffic light system applied at the simulated junctions

Traffic light systems had also been investigated within the project "ORINOKO," a project performed within the German research initiative "Verkehrsmanagement 2010" (Traffic management 2010, 2009). Here, SUMO was used for evaluation of the new weekly signal program plan developed by one of the project partners. This plan is used within a large area around Nürnberg's fair trade center, the "VLS" area. In order to predict the new plan's performance, a very detailed representation of the area was necessary, including both a very detailed network a and demand representation. The network was modeled using NAVTEQ database (2009), and supplemented by number of lanes, design speeds, intersection geometry, and traffic light information. The demand had been computed by a calibration of the VALIDATE (2009) demand data set for this area with available induction loop count data. Additionally, SUMO was extended to simulate weekly program signal plans. This included the implementation of two methods for switching between different programs, named GSP and Stretch.

Due to the relatively small size of the investigated networks, the mentioned projects are not representative for the intended usage of SUMO. The simulation of much larger networks has been performed within the projects "INVENT" and "WJT"/"Soccer"/"DELPHI." While the latter complex of projects is described more deeply in Section 7.7, "INVENT" will be shortly presented now; it shows the intended major application of SUMO: the evaluation of large-scale impacts of traffic management strategies or new technologies.

An example of such evaluations done in “INVENT” (2009) was to determine the effects of different vehicle routing strategies in the case of recognized jams, such as ones caused by oversaturation or temporal access restrictions to parts of the city. Here, the simulation was coupled to a routing module via a socket connection for allowing the simulation to send current travel times to the router and the router to send new routes for the simulated vehicles. The routing module was extended to allow the specification of certain modalities for rerouting vehicles, affecting the frequency of route recomputation (either each x s or in the case of an event), and the information available during the computation of a new route (complete knowledge, knowledge about large delays, additional weights given by a user, road closures). The evaluations were done for large-scale networks for two German cities, Magdeburg and Munich. The initial scenarios were set up by the PTV AG and provided in the form of VISUM and Vissim scenarios. For Munich only the northern highway network was simulated. The Magdeburg scenario consists of a road network which includes major roads and a demand of about 600,000 vehicles for a complete day. SUMO’s ability to deal with large scenarios becomes visible when knowing that a complete day’s simulation of this scenario needs about 45 min using recent PCs.

Another project from the “Verkehrsmanagement 2010” research initiative called “TrafficOnline” (Ehrenpfordt et al., 2006; Höpfner et al., 2006; TrafficOnline, 2009) gave us the opportunity to use SUMO as a platform for evaluation of a GSM-based traffic surveillance system. The simulation’s duty was to reproduce in-vehicle telephony behavior within normal situations for different road network types, such as highways or inner-city areas. Again, the scenarios were set up using networks from a NAVTEQ database and induction loop measures and both the obtained traffic flows and the simulated GSM talk statistics were compared with values from the real world. After their validation, these base scenarios were extended by other influences, such as traffic jams, or other transport modes running parallel to the normal traffic, such as city rail, fast rail, or explicit bus lanes. The purpose was to evaluate (a) the quality of the surveillance under undisturbed conditions, (b) the reaction time of the system in the case of incidents, and (c) how well the surveillance system behaves under different conditions in terms of disturbances by other sources of GSM talks. The evaluations were done by sending the GSM talk statistics gained from the simulation runs to the Institute of Transportation and Urban Engineering (IVS) of the Technical University of Braunschweig, which was the developer of the surveillance system. The IVS used these statistics to predict mean velocities on the road network for intervals of 5 and 15 min. These were sent back to DLR and compared against the travel times obtained from the according simulation.

Two recent projects of external contributors should be mentioned because both built upon own extensions of the simulation module. Honomichl (2008) has investigated attacks on privacy in vehicular communications by extending the simulation by own detectors. Morenz (2008) built a system for predicting public transports' travel times by adapting the simulation state to measured induction loop values online.

As a summary, it should be noted that SUMO has proven to be extensible, not only by its original developers but also by external contributors. This was one of the initial goals which have been achieved. Also, SUMO's capability to simulate large networks has proven to be a great feature for investigation of large-scale effects of new methods or technologies.

7.2 Model Building Principles in SUMO

Because SUMO is not an established software suite, used networks and demands are usually given in the formats used by other simulation packages. Due to this, much work within the development of SUMO had been spent on implementing methods for importing road networks and demand data. What was a need at first got a philosophy over time: the main idea when preparing road networks or demands for a simulation is to have them be generated from digital descriptions and enrich or process them for being usable as input data to the simulation. This is one of the reasons that SUMO does not have a network editor or an integrated editor for the demand, yet (but see Section 7.7). Both preparing the road network and the demands are described in detail in the next sections, followed by a summary on preparing simulation scenarios. Notes on validating a scenario are given in Section 7.5.

7.2.1 Preparing a Road Network to Simulate

When dealing with the simulation of real-life road networks with SUMO, the common approach is to import an available digital road network. SUMO's network importer, a tool called "NETCONVERT," is able to read networks from VISUM, TIGER, ArcView shape files, Vissim, Robocup Rescue League folders, OpenStreetMap, and a "native" XML representation of the road network graph. As one may note, most of the named road networks are not originally designed for being used for a microscopic simulation of traffic. That's why information important for a correct microscopic traffic flow simulation, such as which lane may be used to get to a following road, right-of-way rules on intersections, and even information about traffic light positions or their programs, is often missing.

NETCONVERT tries to help here, by applying heuristics for computing missing values. Some of the heuristics are applied only if certain information is not given. As an example, connections between edges will be determined heuristically for ArcView shape files, which do not contain them. NETCONVERT's heuristics include the following:

- Computation of the turning direction for each road (what is also necessary for later computation of connections between lanes).
- Computation of connections between lanes.
- Computation of intersection types (right-before-left vs. intersections with a high-priority road).
- Computation of right-of-way rules for the roads participating in an intersection.
- Computation of road and intersection geometries.

Additionally, NETCONVERT includes heuristics for determining positions of traffic lights, their programs, and for determining where additional lanes for highway on- and off-ramps should be added.

All in all, the implemented heuristics are a great help when dealing with networks which do not contain the needed information. Usually, one obtains a network which looks realistic with a single call to NETCONVERT. Nonetheless, further, manual inspection and work on the networks are usually needed. This means that in practice, the following procedure must be performed in order to prepare a network for simulation:

- An available digital road network description is imported and written into SUMO-native XML files that describe the road network in terms of nodes (intersections/junctions), edges (roads), and lane-wise connections between them.
- The XML files are converted into a road network that may be used within the simulation.
- The resulting road network is inspected.
- In the case the resulting road network is erroneous, these errors are corrected within the XML files that were used to build it.
- The process iterates until the network topology has a proper quality.
- After the network's topology is fixed, traffic light information must be mapped onto it, mainly by replacing the ones NETCONVERT computes by definitions that are based on signal plans from the real world.

It may be interesting whether all imported formats are used similarly. This should be strongly denied. Within our projects, mainly networks imported from VISUM and from NAVTEQ are used; the second either encoded in a proprietary DLR-internal format or as ArcView shape files. From time to time, the definitions of the road networks to simulate reach us in the form of Vissim simulations. In the past time, we have also put an incremented effort in importing networks from the Open Street Map (OSM, 2009) project in order to obtain real-world examples we are legally able to make public. All other import facilities were mostly implemented for evaluating them for their usability but were not thoroughly tested. Of course, this affects the qualities of the import functions. Formats which were only briefly investigated may contain needful information which is not imported. Also, changes in the format and problems one may encounter if using a certain format get only visible if the format is used frequently. Though no explicit issues are known, it should be mentioned, that the importer for TIGER networks, which are very popular within the vehicle-to-vehicle communication community, is not supported well.

7.2.2 Preparing the Demand

Choosing a method for demand modeling depends on the research topic under consideration, the study area, and the availability of data about this area's traffic. Because SUMO was designed for the simulation of a synthetic population which consists of single persons with distinct routes and explicit departure times, SUMO's "native" demand definition is a list of vehicle departure times with the roads the according trip shall start and end at. Using these definitions, the according complete path through the network can be computed, using a simple shortest path algorithm, yielding in a list of vehicle routes. Such routes through the network – together with their start time and possible additional information about the vehicle – are the input required by the simulation. Additional methods for computing the dynamic user equilibrium by a simulation-based dynamic traffic assignment are available and yield a usable, realistic, and fine grained demand and load description of the study area.

Of course, such "microscopic" demand descriptions are not available at all. Only systems which are based on synthetic populations or agents are able to deliver per-vehicle information about the departure time and this vehicle's trip origination and destination road. Though such systems exist, defining a demand for a certain area is very time consuming, because many sociological data must be gathered and used. This means, that even in our own projects, we always had to "fall back" to common and established descriptions of road traffic demand. For large-scale scenarios covering complete cities, importing origin-destination (O/D) matrices from VISUM or other formats has proven to yield valid results. The import consists of the following two steps, which are both directly supported by tools of the SUMO suite:

- Convert the O/D matrix or matrices into a list of single-vehicle "trip" information, consisting of the departure time and the origin and the destination road.
- Perform a dynamic user assignment in order to obtain a realistic set of routes through the road network.

Such demand descriptions should of course consist of a set of O/D matrices, each defining only a small time window, preferably 1 h or less. O/D matrices which describe a whole day's traffic aggregated into a single O/D matrix are rather too coarse for their direct application in a microscopic traffic simulation. For matrices covering a whole day, SUMO itself allows only to apply a user-defined time line of the given amount's percentages over a day. Processing of multiple O/D matrices is supported directly.

It must be noted that O/D matrices may differ very much in their granularity and how connections between districts and the underlying road network are set up. Though it is possible to map a district's area to the edges it contains and use these as trips' origins and destinations, a valid methodology for this purpose was not yet evaluated or even implemented. The consequence is to use the given connections between districts and the road network. This means in practice that a valuable O/D description should use minor roads of the network as connection to the districts. The contrary – additional connections, which have no counterpart roads in the real world's road network, pointing to major intersections, maybe even ones controlled by traffic lights – should be avoided in any microscopic traffic simulation, because these intersections' attributes and the behavior of vehicles around them diverges strongly from reality.

Fine grained, time-dependent O/D matrices are not always available. Also, in the case of smaller areas, available O/D matrices may be too coarse, because the needed traffic description is wanted to be given on a per-road base, and not by joining roads into districts. For fulfilling this need, the SUMO suite includes two further applications which generate per-vehicle demands. The first one is a router based on turning ratios at intersections. This tool reads definitions of flows and time-dependent turning ratios at intersections. Each flow is described by its starting edge and its amount. Using these data, the tool computes the per-vehicle demand, again as a list of vehicles with their routes and departure times by simply choosing a certain continuation at each junction a vehicle passes according to the given ratios. It should be noted that the procedure is very simple and straightforward, so that its usage is easily set up and the results should be valuable because being easy to evaluate. Note, however, that by using this tool in larger networks is expected to yield weird results, since no guard is build in to hinder the routes from forming loops. Note in addition that this tool is not frequently used at our institution, because collecting turning ratios is only possible for smaller areas, while our work concentrates on large networks. It is not known whether or how often this tool is used by other users.

The second tool uses time series of traffic flow from observation locations. At first, these are classified, marking observation points which have no observation point in front as "sources," and those having no observation point behind as "sinks." Then, streams are inserted at the "source" positions and propagated through the network until they reach a "sink." At each junction at which more than one continuation is possible, the streams are spread in accordance to the following observation points' measures.

This principle is very simple and has the disadvantage that only the streams' distributions across the network are computed, not the vehicles' real routes. Nonetheless, it has proven to be valuable for populating a simulation scenario if only the amounts of vehicles passing observation points must be investigated. The major problem in its usage originates from the need to have all entries and exits of the modeled network to be covered with measures. This is usually not the case. Also, this approach works only for networks where no loops in vehicle routes can occur, making it inappropriate for larger inner-city regions but very well suited for highway corridors.

7.2.3 Summary on Preparing a Simulation Scenario

Following the decision to built road networks fast by importing them, no graphical editor which could support a user by adapting changes manually is available by now. This in fact makes the work for getting a complete road network for large areas very uncomfortable and one should state that the original idea to have the networks imported fast is contradicted partially here by.

The tools for demand generation which are included in the package can deal with standard demand descriptions used within the traffic research and offer further possibilities to generate traffic when real-world data are used. Nonetheless, an additional effort is needed for converting given data for using it as input for these tools.

7.3 Fundamental Core Models

As common to most microscopic simulations, the models for vehicles' longitudinal (car-following) and lateral (lane-changing) behavior are executed separately within SUMO. They interact in a minor manner, as described within the subchapter on lane changing, but a close coupling has not yet been done. At first, the vehicles' lateral movement is computed, their lane-changing in a second step. In the following, both models are described in this order, followed by an outlook on further research.

7.3.1 Longitudinal Vehicle Movement

SUMO uses a modified version of the time-discrete and space-continuous car-following model by Krauß (1998). The model is based on a derivation of a safe gap a vehicle, the EGO, needs to stop behind a leading vehicle, the LEADER, without colliding with him. Both the vehicles' maximum decelerations (assumed to be equal) and EGO's reaction time are considered. By using the usual approximation for the braking distance d(v) = vˆ2/(2b), the following formula for determining a safe speed can be computed. This safe speed, given the distance to the LEADER and the speed of the LEADER, assures a collision-free behavior:

$$v_{\mathrm{safe}}(t) = -\tau \cdot b + \sqrt{(\tau \cdot b)^2 + v_{\mathrm{leader}}(t-1)^2 + 2 \cdot b \cdot g_{\mathrm{leader}}(t-1)}$$

where

$v_{\mathrm{safe}}(t)$ is the safe velocity for time t (in m/s);
τ is EGO's reaction time (in s);
b is the maximum deceleration ability (in m/s^2);

$v_{\text{leader}}(t)$ is LEADER's velocity at time t (in m/s);
$g_{\text{leader}}(t)$ is the gap (between EGO's front and LEADER's back) at time t (in m).

Now, it must be assured that EGO neither accelerates faster than it is able to do nor that it gets faster than its maximum velocity. The result is the "desired" or "wished" velocity, computed as

$$v_{\text{des}}(t) = \min\{v_{\text{safe}}(t), v(t-1) + a, v_{\max}\}$$

where

$v_{\text{des}}(t)$ is the velocity EGO wants to use (in m/s);
$v(t)$ is EGO's current velocity (in m/s);
a is EGO's maximum acceleration ability (in m/s^2);
$v_{\max}$ is EGO's maximum velocity (in m/s).

One major achievement of Krauß' model is to assume a driver is not perfect in realizing the desired speed. Instead, the speed actually chosen is somewhat smaller, and this adds important features to the behavior of the model. For example, this random difference to the desired (optimum) speed leads to spontaneous jams and the slow-to-start characteristic of real drivers. The model implements this driver imperfection by a stochastic deceleration. For obtaining the EGO's next speed, it has to be assured that the vehicle is not moving backward afterward. This makes the last step of computing the vehicle's speed be

$$v(t) = \max\{0, v_{\text{des}}(t) - r \cdot a \cdot \varepsilon\}$$

where

r is a random number, between 0 and 1;
ε is EGO's driver imperfection, between 0 and 1;
$v(t)$ is EGO's final speed for time t (in m/s).

Two extensions have been added to the original model. The first was to apply a decay into the ability to accelerate with increasing speed. As a simplification, this was modeled using a linear function, yielding a speed-dependant acceleration computed as following:

$$a(v) = a\left(1 - \frac{v}{v_{\max}}\right)$$

The second was to reduce the driver imperfection's effects on accelerating from low velocities. Instead of the prior computation of dawdling, a distinction based on the vehicle's speed is done, so that the vehicle's final speed is computed using

$$v(t) = \max\left\{0, v_{\text{dawdle,new}}(t)\right\}$$

where

$$v_{\text{dawdle,new}}(t) = \begin{cases} v_{\text{des}}(t) \cdot \varepsilon \cdot r & \textit{if } \; v_{\text{des}}(t) < a(v_{\text{des}}(t)) \\ v_{\text{veh}}(t) - \varepsilon \cdot r \cdot a(v_{\text{des}}(t)) & \text{otherwise} \end{cases}$$

The model by Stefan Krauß is very fast in execution due to the small number of needed computations and has proved to be valid enough in comparison with other models (see Brockfeld et al., 2004; Brockfeld and Kühne and Wagner, 2004; Brockfeld and Kühne and Wagner, 2005 and Section 7.5). The used driver imperfection has found acceptance as an extension to famous models such as the Wiedemann model (see Brilon et al., 2005).

Nonetheless, having a car-following model is only one part of modeling the longitudinal behavior of a driver-vehicle instance. The original model's applicability for traffic simulation was demonstrated by simulating traffic on a large circular road with the parameters of highway traffic. It is not surprising that when moving to complex scenarios, which include networks with roads of arbitrary lengths, complex right-of-way rules and traffic lights, and different vehicle routes, further methods had to be implemented for making the simulation work at all.

First of all, each vehicle must take into account the infrastructure in front of it. Changes in the speed allowed on the approached roads must be regarded before entering the road. Looking ahead must also be applied in order to follow the right-of-way rules without letting the vehicles decelerate stronger than they are declared to be able to. For assuring a collision-free behavior in networks with a complex infrastructure, the following computations are done:

- Adapt velocity in dependence to LEADER's speed and the distance to him (over the next lanes) as described in the original Krauß model.
- Adapt the speed allowed on the next lane.
- If EGO has no right-of-way on the next intersection, compute two velocities, one for passing the intersection and one based on the assumption that the vehicle has to brake and stop in front of the intersection. Store them. Let the intersection know the vehicle is approaching.
- Continue with the next lane along the route or stop, if the seen lane lengths' sum is larger than the braking distance.

After these steps have been performed for all vehicles, it is decided which vehicles are allowed to move over the intersections by following the intersections' right-of-way rules. The vehicles are moved in accordance to their so computed rights-of-way afterward using the previously stored velocities.

As mentioned, the Krauß model has proven to be valid, usable, and fast. Nonetheless, some issues were noted and should be mentioned:

- The driver imperfection is modeled in a very simple way. It has been not validated against real trajectories and should be assumed to resemble traffic flow's macroscopic behavior, not the (microscopic) behavior of a single driver.
- The simplification done by using a reaction time equal to the model time step of one second is known to be problematic within dense scenarios. Here, often the real flow cannot be reproduced.

7.3.2 Lane-Changing Model

While the longitudinal model has proven to be robust so that no major changes were needed, the lane-changing model is strongly evolving since SUMO's begin. The reason is that the original model formulated by Krauß only incorporates a driver's tactical decisions, mainly formulating a driver's behavior based on the assumption a driver wants to drive fast. The navigational (or strategic) part of choosing a lane – the need to change to a certain lane in order to be able to continue the route – is not regarded.

The currently implemented lane-changing behavior (see also Krajzewicz, 2009, in German) solves the problem by computing a valid path through the network, in the means that the chosen lanes can be used for continuing the route; from now on the term *valid lane* will denote a lane which may be used for continuing the route without the need to change the lane. Each lane of the road EGO is currently at and of the roads following along its route – up to a viewing distance – is examined. Besides the distance EGO may continue using the regarded lane without the need to change to a valid lane, the lanes' occupancies are collected. Given these descriptions of lanes, it is decided for EGO whether a lane change into the direction of a valid lane is needed. This is the case if EGO's distance left to the position from which the route cannot be continued is lower than an assumed distance needed for the lane change. The assumed needed distance is computed using

$$d_{lc,\mathrm{veh}}(t) = \begin{cases} v_{\mathrm{veh}}(t) \cdot \alpha_1 + 2l_{\mathrm{veh}} & \text{if } v_{\mathrm{veh}}(t) \leq v_{\mathrm{swell}} \\ v_{\mathrm{veh}}(t) \cdot \alpha_2 + 2l_{\mathrm{veh}} & \text{otherwise} \end{cases}$$

where

$d_{lc,\mathrm{veh}}(t)$ is the assumed distance vehicle *veh* needs for a lane change in time t (in m);
$v_{\mathrm{veh}}(t)$ is vehicle's *veh* speed at time t (in m/s);
v_{thresh} is a threshold discriminating highway and urban behavior (in m/s, set to 14 m/s);
α_1,α_2 are scaling factors (currently set to 5 and 15 s, respectively);
l_{veh} is vehicle's *veh* length (in m).

The approach takes into account the occupancies of the lanes that must be used until reaching the position where the route cannot be continued, including the current lane, the target lane, and lanes to pass. Therefore, the lengths of vehicles in front of the regarded vehicle are subtracted from the distance left. This forces the simulated vehicles to change the lane at the end of a queue on the destination lane, preventing them from trying to drive beside a jammed lane, first, and then trying to merge into this jammed lane when no further continuation is possible.

For the opposite direction – moving away from a valid lane – similar tests are done. EGO is only allowed to move into the direction if this lane change and the lane changes needed to come back to a usable lane are possible within the distance left, regarding the lanes' occupancies.

For the tactical part of the lane changing – the wish to move forward fast – an approach based on Ehmans (2001) was chosen. During his drive, a driver stores the benefits of changing the lane. The benefit to change a lane is the difference between the safe speeds on the neighbor and on the current lane, computed using the car-following model, and normalized by the maximum velocity the vehicle could use under free-flow condition:

$$b_{l_n}(t) = \frac{(v_{\text{poss}}(t, l_n) - v_{\text{poss}}(t, l_c))}{v_{\max}(l_c)}$$

where

$b_{l_n}(t)$ is the benefit of a vehicle to change to lane l_n at time t;
l_c and l_n are the vehicle's current and neighbor lanes, respectively;
$v_{\text{pos}}(t, l)$ is the velocity the vehicle could drive safe with on lane l at time t (in m/s);
$v_{\max}(l)$ is the maximum velocity the vehicle can take on lane l (free flow, in m/s).

Using the benefits for neighbor lanes, a driver-internal "memory" variable, which represents the simulated driver's wish to change to a neighbor lane, is adapted. If the benefit to change the lane is >0, this benefit is added to this memory, signed by the direction. If the benefit is <0, the current lane is faster than the neighbor lane, the memory value is divided by two, suppressing the wish to change into this direction.

A lane change is initiated if the absolute value of the memory variable is larger than a certain threshold. The sign of the memory variable represents the direction of the lane change. Of course, a lane change is only possible if the lane EGO wants to change to has enough space at EGO's current position. Additionally, the resulting gaps must allow further collision-free continuation of driving.

In the case the situation does not allow EGO to enter the desired lane, he starts to interact with the vehicles which are in front and behind him at the this lane. The vehicle itself and the vehicles at his destination lanes are adapting their speed in dependence of whether they are blocked/blocking at their front or their back using the following rules:

$$v_{\text{next}}(t) = \begin{cases} v_{\text{decel}}(t) & \text{if blocking/blocked at own back and front} \\ v_{\text{decel}}(t) & \text{if blocking/blocked at own front} \\ v_{\text{accel}}(t) & \text{if blocking/blocked at own back} \end{cases}$$

where

$v_{\text{accel}}(t) = v_{cf}(t) + \frac{v_{\max}(t)}{2}$ is the vehicle speed after accelerating (in m/s);
$v_{\text{decel}}(t) = v_{cf}(t) + \frac{v_{\min}(t)}{2}$ is the vehicle speed after decelerating (in m/s);
and

$v_{cf}(t)$ is the car-following speed (including the driver imperfection, in m/s);
$v_{\max}(t)$ is the speed after a maximum possible (in accordance to car-following) acceleration (in m/s);
$v_{\min}(t)$ is the speed after a maximum possible deceleration (in m/s).

The model behaves well for both highway and urban scenarios, assuring the vehicles are choosing their lanes early enough and also assuring that all available lanes are used. Nonetheless, the realized look-ahead along the roads to pass does not consider the behavior of other vehicles. This is problematic as soon as a vehicle's current lane is blocked by standing vehicles, but must be soon used for continuing the route. In these cases, the model tends to suppress the vehicle to change the lane.

7.3.3 Summary on Used Models

The initially implemented models for car-following and lane-changing have evolved by incorporating methods for taking into regard the road infrastructure in front, including the right-of-way rules, and the occupation by other vehicles. They are applicable and valid for most cases. Nonetheless, unwanted behavior was observed for both. To solve these issues, and for allowing further applications, the development will continue. Though its main focus is to allow simulations with time steps <1 s currently, also further work on lane-changing is meant to be done.

7.4 Dynamic Traffic Assignment

The ability to compute a dynamic traffic assignment is an integral need for proper simulation of large area scenarios. SUMO uses an approach developed by Gawron, 1998 where each vehicle (a) has its own route, (b) knows its own travel time through the network, and (c) computes new routes without taking into account the travel times of other vehicles. This algorithm is driven by the travel times the used simulation model computes and not based on assumptions how real drivers choose a route through the network. It converges toward an equilibrium. It is an iterative approach, working as following:

1. Initialize the process by computing the fastest route through the empty network for each vehicle to simulate and add this route to the list of routes known by this vehicle. The probability to use this route by this vehicle is set to 1.
2. Perform the simulation using the currently chosen routes in order to obtain the edges' travel times over simulation time.
3. Compare the mean travel times against those obtained in the previous run (if any) and quit if the algorithm converges, i.e., the mean travel time reduction falls below a given threshold.
4. Compute new routes for all vehicles using the network's current travel times. If a new route for a vehicle was found, add it to the list of routes known by the vehicle. Update all known routes' estimated travel times and their probabilities to be chosen. After that, choose one route for this vehicle taking into account the route choice probabilities and continue with step 2.

In the following, the methods for adapting the route probabilities to the last iteration's travel times are shown. At first, the travel times for the routes known by a driver are adapted to the travel times obtained from the simulation using the following formula:

$$\tau'_d(x) = \begin{cases} \tau_s(x) & x = \text{last chosen route} \\ \beta\tau_r(x) + (1-\beta)\tau_d(x) & \text{otherwise} \end{cases}$$

where

$\tau_d(x)$, $\tau_s(x)$, $\tau_r(x)$ are route x's prior travel times as estimated by driver d, retrieved from the simulation, and reconstructed from the edge travel times that were determined by the simulation, respectively (in s);

$\tau'_d(x)$ is driver d's new estimation of the duration of route x (in s);

β is a factor affecting the speed of adapting remembered travel times to the current.

Using these adapted travel time information, the probabilities to choose one of the known routes are recomputed. The probability for each unused route known by the driver is recomputed by a function that compares its travel time with the travel time of the route used in the last simulation step using the following formula:

$$p'_d(r) = \frac{p_d(r)\ \ (p_d(r) + p_d(s))\ \exp\left(\frac{\alpha\delta_{rs}}{1-\delta_{rs}^2}\right)}{p_d(r)\exp\left(\frac{\alpha\delta_{rs}}{1-\delta_{rs}^2}\right) + p_d(s)}$$

where

$p_d(x)$ is the prior probability to use route x by driver d;
$p'_d(x)$ is the new probability to use route x by driver d;
r is the route used in the last simulation run;
s is another route from the list of known routes;
δ_{rs} is the relative cost differences between routes r and s, computed as

$$\delta_{rs} = \frac{\tau_d(s) - \tau_d(r)}{\tau_d(s) + \tau_d(r)}$$

where

$\tau_d(x)$ is the travel time for driver d to complete route x.

The probability to use the route which was already used in the last iteration step is updated by

$$p'_d(s) = p_d(r) + p_d(s) - p'_d(r)$$

Normally, travel times for edges are collected and aggregated into intervals of 15 min during the simulation's runs. The so obtained time series of edge travel times are then read by the router module and used for the described computation of new routes and probabilities to use known routes. During this process, each edge's travel time for a vehicle's entry time is determined by looking up in the corresponding time series for the interval that matches interval begin $\leq$ entry time < interval end. For α and β, usually values of 0.5 and 0.9 are used, respectively.

The algorithm has proven to generate valuable results. Nonetheless, its iterative nature makes it very slow in execution – in order to get usable assignments, often more than 20 iterations are necessary, each consisting of a computation of new routes and a simulation step. In addition, as vehicles are starting using the fastest routes in an empty network, without an a priori assignment, the first iterations are dominated by large jams, making the simulation additionally slower than the normal execution time.

To solve these problems, several attempts have been undertaken, including implementation and evaluation of macroscopic traffic assignments, and introducing methods which try to solve the problems of slowing down the simulation by jams during the first simulation steps. A report on these methods can be found in Behrisch et al. (2008) and Behrisch et al. (2008). The most promising – and surprising – attempt is the usage of a one-shot assignment. Here, each vehicle is started with its origin and destination edge, the route is then computed at the time the vehicle enters the network. The network's edge weights (travel times) used for computing the currently fastest route are adapted to the situation within the network in each time step using the following formula:

$$w(t,e) = \begin{cases} l(e)/v_{\max}(e) & t = 0 \\ w(t-1,e) \cdot r + l(e)/v_{\text{curr}}(t,e) \cdot (1-r) & \text{otherwise} \end{cases}$$

where

$w(t,e)$ is the weight of edge e at the current simulation step t (in s);
$l(e)$ is the length of edge e (in m);
$v_{\max}(e)$ is the maximum velocity allowed on edge e (in m/s);
$v_{\text{curr}}(t,e)$ is the mean velocity of vehicles on edge e in time step t (in m/s);
r is a remembering factor.

As mentioned, the results of using this approach were surprisingly good, combining a fast execution with short travel times – the measure used for determination whether a network equilibrium is approached – of computed routes. Though, further investigations are still necessary.

7.5 Calibration and Validation

Within the description of the used longitudinal model the need to distinguish between the used car-following model and what the complete simulation does was already mentioned. Because a traffic simulation is a computer application, the validation must be done at different levels, starting at a verification of the computer program as such. This is worth to mention because one can learn from different possibilities to verify a computer application about meaningful verification of models. The current attempt to assure SUMO's correct behavior assumes the following levels:

- Unit Tests
  Unit Tests are very small tests. Each assures that a certain function – the minimal part of a computer application – behaves as should given a set of parameters or given a certain internal state. An example would be to test whether a multiplication function really returns 4 if two parameters, 2 and 2, are given. Of course, most of an application's functions are more complicated. Currently, the usage of unit tests within SUMO is being evaluated, and only a few tests are available by now, written using the googletest framework (2009).
- Acceptance Tests
  An acceptance test compares the output – including what is printed on the command line and the generated files – of a complete application's execution against what was declared to be correct. If the current and the last outputs are same, the test returns a message about a correct behavior of the software, otherwise it reports an error. The SUMO suite is tested each night using more than 2000 acceptance tests, of which almost 800 deal with the simulation itself. The test suite is set up using the "TextTest" (2009) framework.

  The major lesson learned during the work on the tests was to make each test as simple as possible. The verification of large tests, tests including interactions between many vehicles, for example, is time consuming and error-prone. Also, complicated tests are also more sensitive to small changes of the model. The reason is that the generated files are directly compared against each other and a minor change in the output, for example, a difference of a vehicle's speed by 0.01 m/s already sets the test to have failed. This requires the cumbersome verification process to be redone. This is rather not the case if the tests have already been initially set up well defined.

  As a conclusion, it should be stated that acceptance tests are not a proper tool for assuring a simulation's correctness for complex scenarios. Still, they are very valuable for assuring the correct working of an application.

- Model Tests
  Beside the comparison to the fundamental diagram given in Krauß (1998), the car-following model by Krauß was tested within a set of model comparisons together with other microscopic traffic flow models. The reported results (Brockfeld and Wagner, 2004; Brockfeld et al., 2004; Brockfeld et al., 2005) show its applicability to represent real traffic flow and real driver behavior.

The presented possibilities to test a simulation are not yet covering all functions the suite's applications offer. Of course, this will never happen at all, since a complete coverage of possible settings would not be possible due to their infinite number. This means, that a verification of a setup simulation scenario is still necessary. The validation procedure performed within our projects depends mainly on the available data. Normally, induction loop values are available and the simulation is compared against these. This is done by inserting simulated induction loops at the position of the induction loops in the real world. The values generated by these after a simulation's run are directly – despite the normally needed formatting conversion – comparable with real induction loop values. The results are assumed to be valuable because they show whether the modeled flow is correct and is correctly propagated through the road network. In addition, it may be evaluated whether the simulated vehicles' speed matches the reality at the positions the induction loop are placed.

7.6 Extended Modeling Capabilities: Working with External Applications

TraCI, the "*Tra*ffic *C*ontrol *I*nterface," is the contribution done by an external institution of which SUMO benefits most. TraCI extends SUMO by the possibility to interact with a running simulation online by connecting an external application to SUMO using sockets. It was implemented by staff members of the University of Lübeck, mainly Axel Wegener (see Wegener et al., 2008; Wegener 2008). When used, the simulation is triggered from the external application to continue with the next step. This means that in contrary to a "normal" simulation, each step must be explicitly called by the external application. As a result, both the simulation and the external application run synchronously.

TraCI allows asking for attributes of vehicles, traffic lights, induction loops, road infrastructure, and other simulation objects. Using TraCI one can also influence simulated objects. The phase of a traffic light, its duration, or even a complete program of a traffic light can be changed using this interface. It is also possible to control a traffic light completely via this interface, setting explicit states for all signals. TraCI also allows changing a vehicle's maximum speed, forcing it to brake, or to change the lane, give it a new destination or to force a recomputation of a vehicle's route.

The major application of TraCI is to connect SUMO to the communication network simulator ns2 – either directly or via an application in between, such as

TraNS (Piórkowski et al., 2008; EPFL, 2008), which is developed mainly by Michał Piórkowski and Maxim Raya from the EPFL Lausanne. TraNS allows defining V2V applications for their simulation using ns2 and SUMO and is very prominent within the V2V community. The support for TraCI is enabled within SUMO by default since version 0.9.8. Due to being used within the currently running project iTETRIS (see Section 7.7), TraCI's capabilities are ongoing a process of revision and extension.

7.7 Selected Projects, Contribution, and Data

Two projects the DLR is participating in are worth being elaborated more deeply. The first one, DELPHI, is a portal accessible to authorities for managing own reaction forces and the road traffic itself in the case of catastrophe or large event scenarios. The second one is iTETRIS, a project founded by the European Commission which is aimed at establishing a common platform for development and evaluation of traffic management strategies based on V2V/V2I communication. These projects are described in the first two of the following subchapters. Then, a foreign application meant for being used in conjunction with the SUMO package, the "SUMO Traffic Modeler," is described. At last, a large scenario named "TAPAS Cologne" that was made public in the recent time (end of 2008) is introduced.

7.7.1 DELPHI

DELPHI – Deutschlandweite Echtzeit Verkehrs-Lage und Prognose – is the continuation of two former projects, Weltjugendtag 2005 (WJT2005) and Soccer (2006) which gained large public interest. The major scientific challenge was to gather information about the current traffic situation using airborne surveillance systems and to embed these in a simulation-driven representation of the road's traffic state together with conventional induction loop data. The so obtained representation of the real-world traffic was extrapolated half an hour into the future.

DELPHI continues this approach, aiming for (a) a sustainable delivery of the road network's current and future state and (b) offering the authorities to manage their law enforcement and emergency services using this data. Two major German cities are currently covered by the system, Cologne and Munich. DELPHI is web based; it retrieves induction loop values from the local highway administration offices via a dedicated connection. In addition, measures from airborne detectors, developed within the DLR project "ARGOS" and floating car data (Schäfer et al., 2002) are received and included, if available. The system is accessible for a user using an internet browser application.

Besides being shown directly to the user, the traffic information gained from sensors is integrated into a simulation. The simulation itself was targeted at the traffic situation on average weekdays and weekends. This was done by importing networks from NAVTEQ and demands from the VALIDATE data set by PTV AG, first. In further steps, the given demand was assigned to the network, and afterward calibrated to stored induction loop measures using a matrix adaptation approach.

During the system's operation, the simulation is re-started every 5 min, starting to simulate the time 5 min before its execution time. Besides loading the precomputed demand for the time to simulate and the last simulation state, the last induction loop and airborne detector data are used for calibrating the flows and their speeds within the simulation. For the simulation's first 5 min, the collected data is used. The so obtained network state – calibrated to measured values – is stored for the next simulation run. Further 30 min of traffic are then simulated, being additionally calibrated by extrapolated values. Besides extrapolating the state into the future, the simulation also models traffic on roads which are not covered by detectors this way.

This simulation speed – more than six times real time for study areas as large as Munich and Cologne which both have more than one million inhabitants – can no longer be achieved by a pure microscopic simulation. For realizing the system, SUMO was extended by a mesoscopic queue model, originally developed by Gawron (1998), and extended and made more realistic by Eissfeldt (2004). The model has been embedded into SUMO with no change on the interfaces; in order to enable it, SUMO has to be started with only one additional parameter. This allows reusing all available applications from the SUMO package with no change. Note, however, that the mesoscopic extension is not available as a part of the open-source package.

The obtained traffic representation is used by the DELPHI system to allow the user to (a) compute shortest routes, regarding the current traffic state, (b) monitor these routes, and (c) compute isochrones of accessibility. This is enhanced by functionality specifically requested by the authorities which will result in a full-fledged web-based (traffic and event) management tool as the project continues.

7.7.2 iTETRIS

As noted before, SUMO is used often by the communication network simulation community in the context of vehicle-to-vehicle and vehicle-to-infrastructure communication. iTETRIS – an Integrated Wireless and Traffic Platform for Real-Time Traffic Management Solutions – is a project founded by the European Commission aiming at this topic, still with a clear focus on traffic management. The work done in iTETRIS is meant to "... create a long term (beyond the project), global, sustainable, open, vehicular communication and traffic simulation platform facilitating large-scale, accurate, multidimensional evaluation of cooperative ICT solutions for mobility management ..." (iTETRIS, 2009). This goal shall be achieved by work on the simulators themselves – ns3 for networking simulation and SUMO for road traffic simulation – and on the connection between them.

The project builds upon a real city's traffic problems; the situation in the city of Bologna, which administration is one of the project's consortium members, is described and evaluated showing bottlenecks and proposing solutions for solving these. Traffic descriptions for this city's problematic areas are supplied which are then translated into SUMO format from the original Vissim and VISUM sources. Given these evaluations and scenarios, traffic management strategies, which are assumed to be capable to solve the reported problems and which are based on vehicular communication, are derived, implemented, and tested.

A major topic within this project was to assure that the environmental effects of the evaluated strategies will be considered. Therefore, the possibilities to compute gas pollutant (CO, CO_2, NOx, PMx, HC) emissions, fuel consumption, and noise emissions were implemented in SUMO. Gas pollutants emission and fuel consumption were modeled based on HBEFA (INFRAS 2009), a database on vehicular emissions. The values of the HBEFA database were approximated by functions, first. Then, the resulting function parameters were clustered in order to obtain a set of vehicle classes which is smaller than the original one which consists of over 100 classes, in order to ease setting up a simulation scenario. For the noise emission model, Harmonoise (Nota and Barelds and van Maercke and van Leeuwen, 2005) was chosen and implemented. The implementations of both models in SUMO allow computing emissions on per-vehicle, per-lane, and per-road base. The two latter also allow different time aggregation of the values. The implementation of both models is completed and available as an integral part of SUMO since March 2009 (release 0.10.2).

A second major extension of SUMO for its usage within iTETRIS is to allow SUMO to run with time steps smaller than 1 s. This will be achieved by implementing a recently developed car-following model. In addition, the rules for regarding the right-of-way on junctions must be reformulated, because they are currently coupled tightly to the simulation's time step length. Further topics of this project are aiming at extensions toward further possibilities to interact with external applications via TraCI to allow them (a) to control the currently simulated traffic lights, (b) reroute currently simulated vehicles, and (c) allow the simulation of advanced driver assistance systems ADAS based on vehicular communication.

7.7.3 SUMO Traffic Modeler

"SUMO Traffic Modeler" (Papaleontiou and Dikaiakos, 2009) is a graphical editor for traffic demands for a given network. It was written by Leontios Papaleontiou as a part of his Master thesis done at the University of Cyprus. SUMO Traffic Modeler allows loading an existing SUMO-network and to edit "traffic area elements," similar to districts in user assignment tools, graphically. The shape of a traffic area element may be either a polygon or a circle.

Besides defining traffic area elements as such, SUMO Traffic Modeler allows to model demands between them, or use one as a "hot spot" – an area within the road network where vehicles preferably start or end. A further area element type allows to model activity-based demand generation, using a simplified synthetic population approach. Furthermore, demands can be also edited by giving an origin and a destination road. The so generated demands can be exported as per-vehicle trip definitions which can be further processed by tools from the SUMO suite.

The Traffic Modeler is a very interesting application, since it adds a traffic demand modeling tool to SUMO which was missing before. Additionally, it is the first tool which was implemented at an external institution and which is mainly related to questions from traffic modeling and simulation.

7.7.4 TAPAS Cologne

Making the "TAPAS Cologne" scenario available is an approach to supply a high-quality example data set which includes all data needed for performing a simulation and which can be used as a base for own evaluations. The amazing OpenStreetMap (2009) project delivers a free digital road network, but traffic demands are normally not freely available. Data from the TAPAS project (Hertkorn and Kracht, 2002; Hertkorn and Wagner, 2004; Hertkorn), which was already mentioned in Section 7.1, could have been made freely available earlier, because they were generated at the DLR, but the originally used digital road network could not be put in the public domain. After projecting the demand data onto an OSM network, we can now offer a large area covered completely by a normal day's passenger traffic. Applying OSM license, the data is available under the "Creative Commons Attribution-Noncommercial-Share Alike 3.0" license (2009), what means that the data must not be used for commercial purposes, and we have to be notified in the case someone uses it.

The data set in its current form contains (a) the road network imported from the OSM database with no changes, despite applying heuristics for building highway on- and off-ramps and traffic light programs; (b) a set of points of interest and polygons extracted from the OSM database which represent buildings such as shops, hospitals, etc., parks, parking places, and positions one can find public telephones or one can give his bottles back; and (c) the demand of passenger cars for the given road network in form of routes through the network. In addition, the configuration files for the SUMO simulation are given, so that one can execute the scenario out of the box (Fig. 7.1).

The currently given data have a large potential. In combination with the given – and increasingly growing – information about the area in the means of points of interest and polygons, this data set allows to simulate location-based services, planning routes for emergency vehicles, and much more. The availability of information about railways and public transport stops should allow multi-modal simulations.

Note, however, that still a lot of additional work is needed to make this data set really usable. The positions and programs of traffic lights must be revalidated. The same holds for the roads' numbers of lanes and the intersections' right-of-way rules. Public transport lines must be set up as well as delivery traffic, which is not a part of the TAPAS data set.

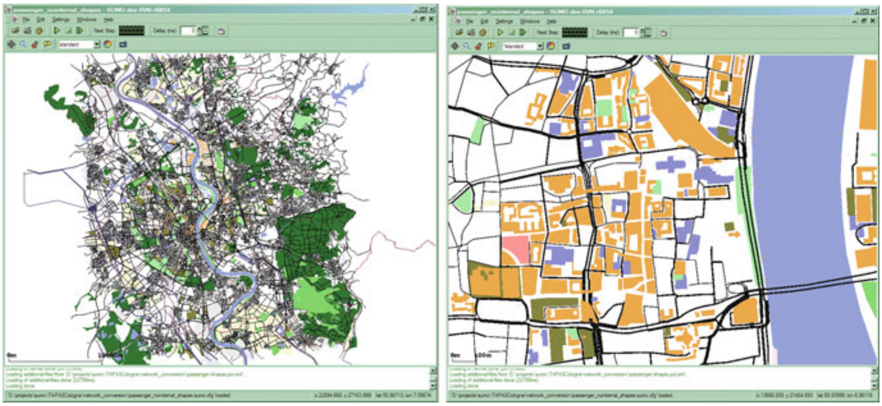

Fig. 7.1 An overview (*left*) and a detail view (*right*) on the TAPASCologne scenario

We hope, that the scenario will be incrementally improved, starting with solutions which are probably working, but not yet completely based on values from the real world. Nonetheless, this work is not meant to be done by DLR only. We hope on the contribution from external participants.

References

Krajzewicz D, Hertkorn G, Rössel C, Wagner P (2002) SUMO (Simulation of Urban MObility) – an open-source traffic simulation. In: Al-Akaidi A (ed) Proceedings of the 4th Middle East symposium on simulation and modelling (MESM20002), S. 183–187, 4th Middle East Symposium on Simulation and Modelling, Sharjah (United Arab Emirates), 2002–09, ISBN 90-77039-09-0

Krajzewicz D, et al (2004) Recent extensions to the open source traffic simulation SUMO. In: Proceedings of the 10th world conference on transport research (on CD), WCTR04 – 10th World Conference on Transport Research, Istanbul, Turkey

SUMO (2001–2009) SUMO web pages. http://sumo.sf.net

GNU.org. Gnu Public License. http://www.gnu.org/licenses/gpl.html. Accessed 9 Apr 2009

Blanke K, Ehling M, Schwarz N (1996) Zeit im Blickfeld: Ergebnisse einer repräsentativen Zeitbudgeterhebung. Band 121 der Reihe Schriftenreihe des Bundesministeriums für Familie, Senioren, Frauen und Jugend. Kohlhammer, Stuttgart, 1996

Hertkorn G, Kracht M (2002) Analysis of a large scale time use survey with respect to travel demand and regional aspects. International association for time use research (IATUR) – conference 2002, Lisbon

Hertkorn G, Wagner P (2004) Travel demand modelling based on time use data. 10th international conference on travel behaviour research, Luzern, 10–15 August 2004

Hertkorn G (2004) "Mikroskopische Modellierung von zeitabhängiger Verkehrsnachfrage und von Verkehrsflussmustern;" Deutsches Zentrum für Luft- und Raumfahrt; Forschungsbericht 2004–29; Köln; http://elib.dlr.de/21014/01/fb_2004-29_v2.pdf

Krajzewicz D et al (2005) Simulation of modern Traffic Lights Control Systems using the open source Traffic Simulation SUMO. In: Krüger J, Lisounkin A, Schreck, G. (eds) Proceedings of the 3rd industrial simulation conference 2005, EUROSIS-ETI, S. 299–302, 3rd Industrial Simulation Conference 2005, Berlin, Germany, ISBN 90-77381-18-X

BMWi. “Verkehrsmanagement 2010” web pages. http://vm2010.de/web/index.php. Accessed 9 Apr 2009
NAVTEQ. NAVTEQ web pages. http://www.navteq.com. Accessed 8 May 2009
PTV AG. PTV validate web pages. http://www.ptv.de/software/verkehrsplanung-verkehrstechnik/software-und-system-solutions/ptv-validate/. Accessed 8 May 2009
INVENT. INVENT project web pages. http://www.invent-online.de/. Accessed 9 Apr 2009
Ehrenpfordt I, Höpfner M, et al (2006) Systematischer Test eines innovativen Verkehrsdatenerfassungssystems. In: Deutsche Messe AG [Hrsg.]: CeBIT in Motion – Forum for Telematics and Navigation, CeBIT in Motion – Forum for Telematics and Navigation, Hannover
Höpfner M, Lemmer K, Ehrenpfordt I (2006) Validation of a GSM-based traffic monitoring system. In: ERTICO [Hrsg.]: ITS world conference 2006, 13th world congress and exhibition on intelligent transport systems and services, London, 8–12 Oct 2006
TrafficOnline. TrafficOnline project web pages. http://www.trafficonline.de/. Accessed 9 Apr 2009
Honomichl C (2008) Analyse und Simulation von Verkettbarkeitsmaßen in pseudonymer, mobiler Kommunikation, Diploma thesis. Technical University Darmstadt. 2008
Morenz T (2008) iTranSIM – Simulation-based vehicle location. Master thesis, University of Dublin
OpenStreetMap. OpenStreetMap web site. http://www.openstreetmap.org/. Accessed 14 Apr 2009
Krauß S (1998) Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics. Hauptabteilung Mobilität und Systemtechnik des DLR Köln. ISSN 1434-8454
Brockfeld E, Kühne R, Wagner P (2004) Calibration and validation of microscopic traffic flow models. In: Transportation research board (ed) TRB 2004 annual meeting, S. 62–70, TRB Annual Meeting, Washington, DC, USA
Brockfeld E, Wagner P (2004) Testing and benchmarking of microscopic traffic flow models. In: Proceedings of the 10th world conference on transport research, S. 775–776, WCTR04–10th World Conference on Transport Research, Istanbul, Turkey
Brockfeld E, Kühne R, Wagner P (2005) Calibration and validation of microscopic traffic flow models. Transportation research records, 1934, S. 179–187
Brilon W, et al (2005) Fortentwicklung und Bereitstellung eines bundeseinheitlichen Simulationsmodells für Bundesautobahnen. Forschung Straßenbau und Straßenverkehrstechnik, Heft 918. Herausgegeben vom Bundesministerium für Verkehr, Bau- und Wohnungswesen
Krajzewicz D (2009) Kombination von taktischen und strategischen Einflüssen in einer mikroskopischen Verkehrsflusssimulation, in Th. Jürgensohn, H. Kolrep (Hrsg.), Fahrermodellierung in Wissenschaft und Wirtschaft, 2. Berliner Fachtagung für Fahrermodellierung, Fortschrittsbericht des VDI in der Reihe 22 (Mensch-Maschine-Systeme), Nr.28, 104–115, VDI-Verlag, 2009, ISBN 978-3-18-302822-1
Ehmans D (2001) Simulationsmodell des menschlichen Spurwechselverhaltens; Simulation model of human lane change behaviour. In: VDI/SAE/JSAE Gemeinschaftstagung, Der Fahrer im 21. Jahrhundert, Berlin 3.-4. Mai 2001.http://www.pelops.de/publikationen
Gawron C (1998) Simulation-based traffic assignment – computing user equilibria in large street networks. PhD Dissertation, University of Köln, Germany
Behrisch M, Krajzewicz D, Wagner P, Wang Y-P (2008) Comparison of methods for increasing the performance of a DUA computation. In: Proceedings of DTA2008, DTA2008 international symposium on dynamic traffic assignment, Leuven, Belgium
Behrisch M, Krajzewicz D, Wang Y-P (2008) Comparing performance and quality of traffic assignment techniques for microscopic road traffic simulations. In: Proceedings of DTA2008, DTA2008 international symposium on dynamic traffic assignment, Leuven, Belgium
googletest. googletest web site. http://code.google.com/p/googletest/. Accessed 14 Apr 2009
TextTest. TextTest web site. http://texttest.carmen.se/. Accessed 9 Apr 2009
Wegener A, Piórkowski M, et al (2008) TraCI: an interface for coupling road traffic and network simulators. In: Proceedings of the 11th communications and networking simulation symposium (CNS′08), Ottawa, Canada

Wegener A, Hellbrück M, et al (2008) VANET simulation environment with feedback loop and its application to traffic light assistance. In: Proceedings of the 3rd IEEE workshop on automotive networking and applications, New Orleans, LA, USA

Piórkowski M, Raya M, et al (2008) TraNS: realistic joint traffic and network simulator for VANETs. SIGMOBILE Mob Comput Commun Rev 12(1):31–33. ISSN 1559-1662. doi: http://doi.acm.org/10.1145/1374512.1374522

EPFL (2008) Trans web site. http://trans.epfl.ch Accessed 05 July 2010

Schäfer R-P, Thiessenhusen K-U, Brockfeld E, Wagner P (2002) A traffic information system by means of real-time floating-car data. In: ITS world congress, 11–14 Oct 2002

Eissfeldt N (2004) Vehicle-based modelling of traffic. PhD thesis, University of Cologne, Cologne, Germany

iTETRIS. iTETRIS web site. http://www.ic-itetris.eu/. Accessed 14 Apr 2009

INFRAS. Handbook emission factors for road transport. http://www.hbefa.net/. Accessed 14 Apr 2009

Nota R, Barelds R, van Maercke D, van Leeuwen H (2005) Harmonoise WP 3 – Engineering method for road traffic and railway noise after validation and fine-tuning. Harmonoise Technical Report HAR32TR-040922-DGMR20 (Deliverable 18)

Papaleontiou L, Dikaiakos M. D. (2009) TrafficModeler: A Graphical Tool for Programming Microscopic Traffic Simulators through High-level Abstractions. IEEE 69th Vehicular Technical Conference (VTC2009-Spring), April 26–29, 2009. Barcelona, Spain

Creative Commons. Attribution-Noncommercial-Share Alike 3.0 license web site. http://creativecommons.org/licenses/by-nc-sa/3.0/. Accessed 14 Apr 2009