SUMO (Simulation of Urban MObility)
An open-source traffic simulation

Daniel Krajzewicz, Georg Hertkorn and Peter Wagner
German Aerospace Centre,
Institute for Transportation Research
Rutherfordstr. 2, 12489 Berlin, Germany
E-mail: Daniel.Krajzewicz@dlr.de,
Georg.Hertkorn@dlr.de, Peter.Wagner@dlr.de

Christian Rössel
Centre for Applied Informatics Cologne
Weyertal 80, 50931 Köln, Germany
E-mail: roessel@zpr.uni-koeln.de

KEYWORDS
traffic simulation, microscopic, continuous, multimodal, open source, car-driver model, traffic research, road traffic

ABSTRACT

As no exact model of traffic flow exists due to its high complexity and chaotic organisation, researchers mainly try to predict traffic using simulations.
Within this field, many simulation packages exist and differ in their software architecture paradigm as well as in the models that describe traffic itself.
We will introduce yet another system which, in contrast to most of the other simulation software packages, is available as an open-source program and may therefore be extended in order to fit a researcher's own needs and also be used as a reference testbed for new traffic models.

INTRODUCTION

When trying to improve traffic, a valid model to work with is needed.
Although some people may assume traffic can be described by departure times and routes with certain durations, traffic is highly conditioned by an individual's private wish for mobility – making up around 60% of traffic – and due to this, neither departure times nor fixed and earlier known routes are available. This is a great problem for modelling traffic itself. Especially the private transit leads to an impossibility of describing traffic by the use of mathematical formulas. Both, a modern human being's wishes to leave and arrive at certain places and at certain times on the one hand, and the movement of the vehicle on the street on the other, influence traffic and one another: the street network's work load depends on the drivers' departure times and determines the speed of movement. Vice versa the load affects the departure times of the drivers' as they wish to move fast and arrive at a certain time.
Beside this, traffic is conditioned by values like the weather, the infrastructure within the region or other incidents affecting the system.
This complexity yields in varied behaviour of the whole system where system means the generation of traffic and traffic itself, and as no valid mathematical models that take into account all these influences are available, simulation is the only way to show weak points of the street network or predict its traffic.
For this purpose, many simulation software packages were developed. Some of them were tested within the SMARTEST-project (SMARTEST 1999).
Such traffic simulation software packages differ in their portfolios of modelled artifacts as well as in their usage paradigm: Some are conceived as Windows-using applications while other which are rather tools for traffic researchers, are simple command-line tools or programs. SMARTEST also holds a list of features that such packages have.
SMARTEST was mostly dealing with microscopic simulations. Simulations of this type regard single vehicles as atomic parts, not the whole traffic flow itself (macroscopic) or single parts of the vehicles or drivers (sub-microscopic). Such vehicle models may be discrete in time and space using cellular automata or only discrete in time (Nagel and Schreckenberg 1992, Brockfeld et al 2001) or may even be fully continuous models. Figure 1 shows the difference between space-discrete and space-continuous simulations.

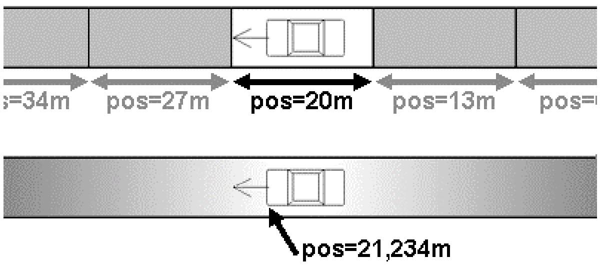

Figure 1: Space-discrete vs. Space-continuous simulation

From a researcher's point of view, when using the available simulation software packages, several problems may arise due to their availability as a ready-to-use software black-box, especially when commercial products are regarded.
At first, one can not examine the underlying model of a simulation. Also, due to different software architectures, a comparison of different models' features (like simulation speed, its ability to describe the reality, etc.) is difficult if not impossible. Furthermore, such simulation tools can not be spontaneously extended by introducing ones own ideas such as new types of sensors, measurements or models.

To introduce a tool which accomplishes these not yet supported tasks, our institute – together with the Centre for Applied Informatics (Cologne, Germany) – is working on a traffic simulation software called SUMO („Simulation of Urban MObility"). In fact, this software is a continuous, microscopic and multi-modal traffic simulation and is – in spite of it's name - also capable of modelling traffic on networks larger than single cities, e.g. highway networks, without any changes.

THE SIMULATION

Basic Paradigms

SUMO is conceived to simulate a traffic road network of the size of a city.
As the simulation is multi-modal, which means that not only are car movements within the city modelled, but also public transport systems on the street network, including alternative train networks, the atomic part of the simulation is a single human being. This human being is described by a departure time and the route he/she takes which again is made up of subroutes that describe a single traffic modality.

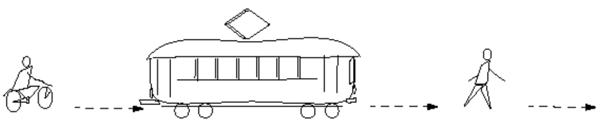

Figure 2: Multimodality

Thus, a simulated person may take his/her car to the nearest public transportation system station and continue his travel by other means of transport. Apart from movements using motorised vehicles, a person may also walk. Walking is not simulated at all but is modelled estimating the time the person needs to reach the destination. Figure 2 displays a such a compound route.

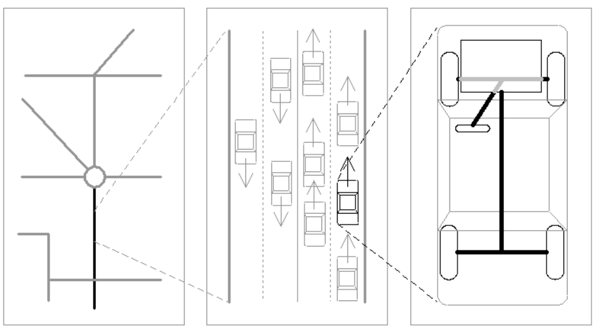

Figure 3: Different simulation classes (from left to right: macroscopic, microscopic, sub-microscopic)

The traffic flow is simulated microscopically. This means, that every vehicle that moves within the simulated network is modelled individually and has a certain place and speed. In every time step which has a duration of 1s, these values are updated in dependence to the vehicle ahead and the street network the vehicle is moving on. The simulation of street vehicles is time-discrete and space-continuous. As our car-driver model is continuous - as the majority of car-driver models are - we decided to use this approach.
When simulating traffic, the street attributes, such as maximum velocity and right of way rules, are regarded.

Features

In the current version – 0.7 – SUMO contains the following features:

- collision free vehicle movement
- different vehicle types
- multi-lane streets with lane changing
- junction-based right-of-way rules (junctions with streets having equal / different priorities, e.g. right-before-left)
- lane-to-lane connections
- a XML-raw-output containing information about the state of the net for every time step
- detectors with independent GnuPlot or CSV (comma separated value) – output
- input from XML-files which may be spread over several files for a better handling

Car-Driver Model

The model used currently within SUMO is the Gipps-model extension (invented and described in: Krauß 1998, Janz 1998). It is capable of displaying main features of traffic like free and congested flow.
In each time step the vehicle's speed is adapted to the speed of the leading vehicle in a way that yields to a collision-free system behaviour within the following simulation step(s).
This velocity is called the safe velocity vsafe, and is computed using the following equation:

$$v_{safe}(t) = v_l(t) + \frac{g(t) - v_l(t)\tau}{\frac{\bar{v}}{b(\bar{v})} + \tau}$$

$v_l(t)$: speed of the leading vehicle in time t

$g(t)$: gap to the leading vehicle in time t

tau: the driver's reaction time (usually 1s)

b: the deceleration function

To bind the acceleration to the vehicle's physical abilities, the resulting "wished" or "desired" speed is computed as the minimum of the vehicle's possible maximum velocity, the vehicle's speed plus the maximum acceleration with the safe velocity computed as shown above, therefore a vehicle will not "drive" or "accelerate" faster than is possible for it:

$$v_{des}(t) = min[v_{safe}(t), v(t) + a, v_{max}]$$

Further, the driver is simulated by assuming he is making errors and so fails to perfectly adapt to the desired velocity. This is done by subtracting a random "human error" from the desired speed:

$$v(t) = max[0, rand[v_{des}(t) - \epsilon a, v_{des}(t)]]$$

As the vehicle must not drive backwards, once again – after the previous computations – the maximum of the computed speed and zero must be taken and will be the vehicle's final speed for the current time step.
This model is collision-free to allow simulations without any artifacts that arise from the imperfection of the underlying model (An impressive in-depth description of the model and the underlying assumptions and rules may be found in Krauß 1998 and Janz 1998).

Traffic Lights

Traffic lights play an important role within the traffic management as they improve traffic flow. Apart from simple right-of-way rules, each simulated junction may also be a junction with traffic lights. As some junctions in Germany allow to ignore the red stoplight when turning right, an extension to the right-of-way rules regarding this is being implemented.

Simulation Output

By now, two different outputs are available. The first is a so-called „raw" output which contains all edges (streets) and all lanes along with the vehicles driving on them for every time step, where vehicles are described by their name, position and speed. This output is complete and may be used as input to post-processing tools for evaluation. However, a large simulation produces a nearly unmanageable amount of data, so other outputs have been invented.
The first processed output that may be generated by the simulation, is a log-file made by simulated detectors which may be positioned on a certain position of a certain lane. These detectors are a simulation of induct loops and are able to compute the flow, the average velocity on the lane, and other values. The results of this computation are written into a file using the CSV or the GnuPlot-format. Every detector has it's own file.

Software Development

Being a tool for traffic researchers, SUMO is designed to be fast and exact instead of trying to be a software that is pleasant to look at. So, although the implementation of a GUI will be one of our next tasks, the main program is meant to be started from a command line and produce an output which must be post-processed when one wants visual results. This prevents data arising from the GUI from slowing down the system, giving more memory and system time to the simulation itself.
SUMO is implemented in C++. During development, we try to use only standardised parts of this language. One of them is the STL which – when not coming directly with the compiler – may be additionally downloaded as free implementations exist (STLPort 2001). The code is well documented and formatted and we will follow Ellemtel-guidelines (Ellemtel 1990-1992) that assure portability compatibility with most systems.
Due to this, our software is compilable using most platforms and we validated this for the following environments:

- Windows using MSVC++
- Solaris using SUN-C++-compiler and STL-Port
- Linux using gcc

Simulation Benchmarks

The simulation is capable of simulating large cities like Berlin, Munich or Cologne on a normal desktop computer. Most of our tests were done on Intel PC's running at 933MHz and having 256MB of memory.
The simulation is capable of simulating around 1Million vehicle movements per second depending on the network's complexity. Further optimisations will follow.

Extensibility

Hoping for the participation of other interested persons, we try to supply potential developers with all information needed to extend and modify the program. Beside things like intelligent traffic lights, models for cars equipped with ACC systems, etc., we hope to supply interested researchers with a common testbed for their microscopic models.
The documentation (SUMO 2002) shows how to extend the simulation – and other tools.

ONGOING PROJECTS

Within our institute some projects use SUMO to validate their assumptions about new technologies. The following projects have already started using SUMO or will use it within the next months:

- A project from California, investigates whether the detector loops spread on different highway lanes may be used to predict jams.
- We try to validate our traffic flow predictions based on floating car data retrieved from Taxis in Berlin, Vienna and Würzburg .
- An internal project is trying to predict the advantages of new sensor technologies, on route prediction and the concluding traffic light phases.
- A project from the US tries to improve traffic within a highway off-ramp area.

ADDITIONAL TOOLS

SUMO consists of more than a single application. Some other modules exist that allow to build assigned data needed for simulations and research.
The following modules are now being developed:

Sumo-netconvert

Due to it's high complexity, the SUMO's network description is not meant to be generated by a human user. Instead, we use this tool to convert common data like lists of edges and optional nodes into a complete SUMO-network.

During this process, SUMO-NETCONVERT reads in the available data, computes the needed input for SUMO and writes the results into a XML-file.

By now, four different input formats may be converted into SUMO-networks:

- simple XML-data containing edge types, nodes and edges
- CSV-data containing edge types, nodes and edges
- Cell-input files (Cell is a queue-simulation developed by the ZAIK)
- VISUM-networks

As SUMO is used within the INVENT-project, some further import functions will be implemented within this year: ArcView, VISSIM and possibly GDF where an indirect import of GDF is already possible by converting it into XML using a script and then using the generated XML-descriptions as input to SUMO-NETCONVERT.

The next figure shows which data may be computed from a simple list of nodes and edges. The flow is aligned from left to right and from top to bottom. The first step is to determine the priorities on the junction, the second to compute the relationships between the lanes and edges that may be reached, and within the third step, the destination edges are split among the incoming lanes. The computation is flexible and depends on the number of incoming and outgoing edges and their sizes as well as the priorities they have within the network and on the resulting type of junction.

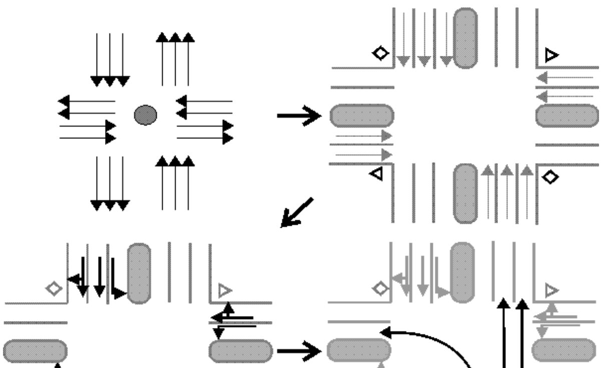

Figure 4: Conversion of simple, plain network data into a complete description

The next picture shows the need of such a net-building tool when a whole city is wished to be modelled. Neither the number of the streets nor the complexity of their relationships allows the processing of the network description by a human user.

SUMO-NETCONVERT is also responsible for the building of traffic light phases where either the priorities of the streets that build up a certain junction or their traffic flows are used. As in the real world – at least in Germany – phases of traffic are adapted individually for each junction and the certain phase data may not be available for all junctions, we use heuristics to generate a realistic output.

Figure 5: The city of Berlin build from high-quality digital map data

Sumo-router

Beside the static part – the network – the simulation consists of moving vehicles. With the increase of the quality of simulations, the need to model a populations' mobility has increased as well. In such cases, vehicles are not spread statistically over the network, instead a single person's daily plan consisting of routes with certain departure times is used. While data needed to describe the departure times and a route's origin and destination are given, the routes themselves must be computed. To avoid online-computation of these during the simulation, this computation is done using a separate module, the SUMO-ROUTER. This module reads the departure times, origins and destinations for a set of virtual humans that will be simulated, then computes the routes through the network itself using the well-known Dijkstra routing algorithm (Dijkstra 1959).

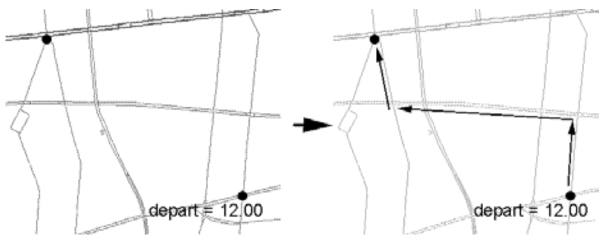

Figure 6: Routing in a small network

As the speed on the streets changes with the traffic amount and therefore the computation of routes using a network where the traffic is not yet known does not

regard the real-world situation, the routing will be done using the Dynamic User Equilibrium approach developed by Christian Gawron (Gawron 1998) where routing and simulation are repeated several times to achieve a real-world behaviour of drivers.
Furthermore, the router supports dynamic network load: the fact that the load on a depends on the time of day is also regarded.

Planned

Other tools are planned due to the growing number of different projects investigating varoius topics and methods.
Some planned modules are:
- traffic light optimisation
- post-processing tools for raw-output evaluation

CONCLUSIONS

We try to build and establish a common platform for traffic research by providing a simulation tool that may be applied by non-programming users in a simple way supporting them with methods and tools mostly needed when working on traffic simulations. Due to it's high portability, the tool may be used on different operating systems.
Besides this, the platform is also extendable by others in order to allow them to improve the software and include ideas we have not taken into account.
Additionally, models originally implemented may be replaced by own methods allowing their comparison to the existing models in respect to the simulation quality and speed.

FUTURE WORK

Ongoing Work

Further work will be done as this simulation will be used by our institute for different purposes. Projects concerned about the verification of floating car data, the prediction of improvements on traffic foreseeing and traffic light optimisation through new sensors that observe traffic will increase the need to extend the simulation software by artifacts reflecting these devices and so will increase the simulation's complexity.
To trace and validate the simulation, a GUI will be implemented where the loaded map, together with vehicle movements and generalised information, will be displayed. To stay portable, the GUI will be implemented using the Qt-windowing library (TrollTech 2002) which is free to use and available for most systems including Windows, UNIX/Linux and Macintosh. Two different types of GUI support will be implemented. While the first application is the simulation itself extended by a windowing system, the second will be an interface that soley displays data coming from simulation module(s).
Some running projects have shown the need to integrate the software into other software packages or to build interfaces between the software and other programming languages. This, too, will be investigated.
Also, as the amount of traffic to be simulated is growing and some research experiments need several simulations to compute a single value (for instance in case of traffic light optimisation where several timing schemes must be tested), the system will be extended to allow the usage of computer clusters.

Participation

We also hope to gain help from other persons or institutes that are interested in traffic simulation and want to participate on SUMO's development by extending, improving or simply using and criticizing it. We highly invite you to visit the SUMO-pages at http://sumo.sourceforge.net.

REFERENCES

Brockfeld, E. et al. 2001. „Optimizing Traffic Lights in a Cellular Automaton Model for City Traffic“. In: Physical Review E 64, 056132

E. W. Dijkstra. 1959. „A note on two problems in connexion with graphs“. In: Numerische Mathematik, 1:269-271.

Directorate General VII - Transport of the European Commission. „SMARTEST home page“. http://www.its.leeds.ac.uk/projects/smartest/

Ellemtel. 1990-1992. Coding Standards Page. http://membres.lycos.fr/pierret/cpp2.htm

Christian Gawron. 1998. „Simulation-Based Traffic Assignment“. Inaugural Dissertation.

Stefan Janz. 1998. „Mikroskopische Minimalmodelle des Straßenverkehrs“. Diploma Thesis.

Stefan Krauß. 1998. „Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics“. Hauptabteilung Mobilität und Systemtechnik des DLR Köln. ISSN 1434-8454

K. Nagel, M. Schreckenberg. 1992. Journal of Physics I 2, 2221

STLPort. 2001. Company Homepage. http://www.stlport.org/index.html

SUMO: G. Hertkorn, D. Krajzewicz, C. Rössel. 2002. SUMO Homepage. http://sumo.sourceforge.net

TrollTech. 2002. Troll Tech Homepage. http://www.trolltech.com

BIOGRAPHY

Born in Bydgoszcz, Poland, 1972, **DANIEL KRAJZEWICZ** has finished his study of computer science at the Technical University in Berlin by the middle of the year 2000 with artificial intelligence and computer graphics as main topics. After work on text classification he changed to the Institute for Transportation Research of the German Aerospace Centre where he now works on a cognitive driver model and an open-source urban traffic simulation.