# OLSIM: A New Generation of Traffic Information Systems

Roland Chrobok, Sigurður F. Hafstein, Andreas Pottmeier

University Duisburg-Essen

*Summary*

Detailed and reliable information about the current traffic state is hardly obtainable by the road user. Therefore, we propose a web based visualization of the current and future traffic load of the autobahn network of North Rhine-Westphalia, Germany. This novel traffic information system called OLSIM is based on an efficient and highly realistic traffic flow model, which is fed by traffic data of 4,000 detecting devices across the road network every minute, and a graphical user interface which can be accessed at *www.autobahn.nrw.de*.

## 1. Introduction

Since the construction of the first autobahn in Germany in the early 30th between Bonn and Cologne, the vehicular traffic has risen in a dramatic manner, especially in North Rhine-Westphalia. Whereas at first the autobahns could handle the traffic demand easily, nowadays, particularly in densely populated regions, the existing autobahn network has reached its capacity limit. The daily occurring traffic jams cause significant economic damage. Moreover, in these areas, it is usually hardly possible and socially untenable to enlarge the existing network. This is in particular true for the German state of North Rhine-Westphalia. The network is not able to cope with the demand in the daily rush hours and the drivers have to deal with the traffic jams in the Rhine-Ruhr region (Dortmund, Duisburg, Essen, Krefeld, Düsseldorf, etc.) and around Cologne (Leverkusen, Neuss, etc.). The prognosis for the future paints an even worse picture as the demand will increase further. New information systems and traffic management concepts are thus truly needed.

Therefore, we established the advanced traffic information system OLSIM which gives the internet user the opportunity to get information about the current traffic state, a 30, and a 60 minute prognosis of the autobahn network of North Rhine-Westphalia. Our approach to generate the traffic state in the whole autobahn network is to use locally measured traffic data, mainly provided by about 4,000 loop detectors as the input into an advanced cellular automaton traffic simulator. These measured data, which are delivered minute by minute, include especially the number of vehicles and trucks passed, the average speed of the passenger cars and trucks, and the occupancy, i.e., the sum of the times a vehicle covers the loop detector. The simulator does not only deliver information about the traffic states in regions not covered by measurement, but also gives reasonable estimates for other valuable quantities like travel times for routes, a quantity that is not directly accessible from the measurements of the detectors. As a further improvement we combine the current traffic data and heuristics of aggregated and classified traffic data to forecast the future traffic state. In the first step we gave a short-term forecast for 30 minutes, which was extended in the next step by a 60 minute prognosis. This information is completed by the temporal and spatial road work and actual road closures. All these valuable traffic information is integrated in a Java applet that can be accessed by every internet user at *www.autobahn.nrw.de.*

## 2. General Concept of the Traffic Information System OLSIM

The intention in developing the traffic information system OLSIM is to offer the opportunity to inform the road user fast and efficient about the current and the predictive traffic state. Therefore, the information mentioned above has to be collected and prepared in a manner that is useful for the user. The general setup of the traffic information system OLSIM is depicted in Fig. 1.

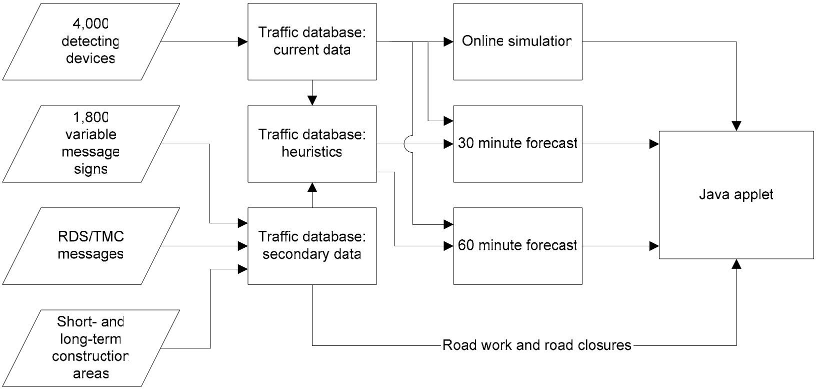


*Fig. 1: The architecture of the traffic information system OLSIM.*

First of all the different kinds of data have to be collected. Especially, the traffic data is stored in a database. These are sent from 4,000 loop detectors to the central OLSIM server every minute. The same holds for the data of the control states of about 1,800 variable message signs (VMS) that are located across the network. Furthermore, the data of road works are sent from the traffic centrals to OLSIM. The messages of short term construction areas are sent daily, those of permanent construction areas every two weeks. The data include the location and the duration of the construction area and an estimate whether the construction area will cause congestion or not.

Another data source are the so called RDS/TMC-messages. These messages are information provided by the traffic warning service and include all kind of warnings concerning the current traffic like traffic jams, accidents, road closures, and reroutings. These data are sent to the OLSIM server immediately when they are generated.

To generate a valid picture of the traffic state many kinds of data fusion techniques are needed. First, the actual traffic data are integrated into the microscopic traffic simulator. Using it, every vehicle that is measured at any time at one of the 4,000 loop detectors is directly fed into the simulation and virtually moves on. In this way the point information of the loop detectors is merged into a network wide traffic state. Such simulations are running for the current traffic state, for the 30 minutes, and for the 60 minutes forecast. In contrast to the online simulation, the forecasts are based on a combination of the actual traffic data and heuristics that are frequently generated and stored in a second database.

These heuristic traffic patterns are aggregated data which are classified in different days (work days, holidays, etc.) and secondary data like road constructions, variable message signs, and special events.

The second level of data fusion is done in the java applet at the website *www.autobahn.nrw.de*. Traffic state, construction areas, and road closures are integrated in one graphical user interface. Here each section is colored according to its calculated traffic state. Moreover, the construction areas and the road closures are marked in the map at their location. Their temporal parameters are shown in the status bar. The user can easily choose between the current traffic situation, the 30, and the 60 minute prognosis.

The microscopic traffic simulation, on which the core of the information system is based, is focused on in the next sections. So, in the following the traffic simulator, the topology, and some special problems which arise when such a complex network is mapped in the computer are explained in detail.

## 3. Simulation Model

The kernel of the online simulation is an advanced and highly realistic traffic simulation model. Because the data is fed into the simulator and processed by it every minute it has to be at least real-time. Due to their design cellular automata models are very efficient in large-scale network simulations (5, 11, 20, 22, 24). The first cellular automaton model for traffic flow that was able to reproduce some characteristics of real traffic like jam formation was suggested by Nagel and Schreckenberg (18) in 1992. Their model has been continuously refined in the last 10 years. The model we implemented in our simulator uses smaller cells in comparison with the original Nagel-Schreckenberg model, a slow-to-start rule, anticipation, and brake lights. With these extensions the cellular automaton traffic model is able to reproduce all empirically observed traffic states. Further, we use two classes of different vehicles, passenger cars and trucks, where the trucks have a lower maximum velocity and different lane changing rules.

Smaller cells allow for a more realistic acceleration and more speed bins. Currently an elementary cell size of 1.5 m is used, in contrast to the 7.5 m in the original Nagel-Schreckenberg model. A vehicle occupies 2-5 consequent cells. This corresponds to speed bins of 5.4 km/h and an acceleration of 1.5 $m/s^2$ (0-100 km/h in 19 s), which is of the same order as the "comfortable" acceleration of about 1 $m/s^2$. By using velocity dependent randomization (1), realized through the introduction of 'slow-to-start rules', meta stable traffic flows can be reproduced in the simulation, a phenomenon observed in empirical studies of real traffic data (7, 12, 25). The inclusion of anticipation and brake lights (2, 15) in the modeling leads to a more realistic driving, i.e., the cars no longer determine their velocity solely in dependency of the distance to the next car in front, but also take regard to its speed and whether it is reducing its speed or not.

In the Nagel-Schreckenberg model there is only one global parameter, the probability constant (or dawdling parameter) *p*, and every vehicle, say vehicle *n*, is completely determined by two parameters: its position $x_n(t)$ and its velocity $v_n(t) \in \{0, 1, \dots, v_{max}\}$ at time *t*. When the vehicle *n* decides in the time-step $t \bullet t+1$ how fast it should drive, it does this by considering the distance $d_{n,m}(t)$, i.e., the number of empty cells, to the next vehicle *m* in front. The modifications mentioned above of the Nagel-Schreckenberg model imply that we have to add some new parameters to the model. When the simulation algorithm decides whether a vehicle *n* should brake or not it does not only consider the distance to the next vehicle *m* in front, but estimates how far the vehicle *m* will move during this time-step (anticipation). Note, that the moves are done in parallel, so the model remains free of collision. This leads to the effective gap

$$d_{n,m}^{eff}(t) := d_{n,m}(t) + \max(v_m^{\min}(t) - d_s, 0)$$

seen by vehicle *n* at time *t*. In this formula $d_s$ is a safety distance and

$$v_m^{\min}(t) := \min(d_{m,l}(t), v_m(t)) - 1$$

is a lower bound of how far the vehicle *m* will move during this time-step. $d_{m,l}(t)$ is the number of free cells between car *m* and car *l* in front. Brake lights are further components of the anticipating driving. They allow drivers to react to disturbances in front earlier by adjusting their speed. The variable $b_n(t) = on$ if car *n* has its brake lights on and $b_n(t) = off$ if they are off.

Several empirical observations suggest that drivers react in a temporal- rather than a spatial-horizon (6, 17). For this reason the velocity-dependent temporal interaction horizon

$$t_n^s(t) := \min(v_n(t), h)$$

is introduced in the model. The constant *h* determines the temporal range of interaction with the brake light $b_m(t)$ of the car *m* ahead. Car *n* does only react to $b_m(t)$ if the time to reach the back of car *m*, assuming constant velocity ($v_n = const.$) and car *m* standing still, is less than $t_n^s(t)$, i.e.,

$$t_{n,m}^h(t) := \frac{d_{n,m}(t)}{v_n(t)} < t_n^s(t).$$

The estimations for $h$ vary from 6 s (6), 8 s (17), 9 s (10) to 11 s (4). Another estimation can be obtained from the analysis of the perception sight distance. In (21) velocity-dependent perception sight distances are presented that, for velocities up to 128 km/h, are larger than 9 s. Therefore $h$ is set to 6 s as a lower bound for the time headway (16).

The third modification of the Nagel-Schreckenberg model implemented in the simulator is a velocity dependent randomization, which means that the probability constant $p$ is replaced with a probability function dependent on the velocity of the vehicle. Further, the probability is also a function of the brake light of the next vehicle in front. In every time-step for every vehicle $n$ with vehicle $m$ next in front, the probability that the vehicle $n$ brakes is

$$p = p(v_n(t), b_m(t)) := \begin{cases} p_b, & \text{if } b_m(t) = on \text{ and } t^h_{n,m}(t) < t^s_n(t), \\ p_0, & \text{if } v_n(t) = 0, \\ p_d, & \text{default.} \end{cases}$$

The parameter $p_0$ tunes the upstream velocity of a wide moving jam and $p_d$ controls the strength of the fluctuations.

With this parameter set the model is calibrated to the empirical data. The best agreement can be achieved for $d_s$ = 7 cells, $h$ = 6, $p_b$ = 0.96, $p_0$ = 0.5, and $p_d$ = 0.1. For a detailed analysis of the parameter set see (16).

To sum up, to move the vehicles forward in the network the algorithm executes the following steps in parallel for all vehicles $n$:

### *3.1. Move forward (drive):*

– Step 0: Initialization:

For car $n$ find the next car $m$ in front. Set $p_n(t) := p(v_n(t), b_m(t))$ and $b_n(t+1) :=$ *off*.

– Step 1: Acceleration:

$$v_n(t+\frac{1}{3}) := \begin{cases} v_n(t), & \text{if } b_n(t) = on \text{ or } (b_m(t) = on \text{ and } t^h_n(t) < t^s_n(t)), \\ \min(v_n(t)+1, v_{\max}), & \text{default.} \end{cases}$$

– Step 2: Braking:

$$v_n(t+\frac{2}{3}) := \min(v_n(t+\frac{1}{3}), d^{eff}_{n.m}(t)).$$

Turn brake light on if appropriate:

$$\text{if } v_n(t+\frac{2}{3}) < v_n(t), \quad \text{then } b_n(t+1) := on.$$

– Step 3: Randomization with probability $p_n(t)$:

$$v_n(t+1) := \begin{cases} \max(v_n(t+\frac{2}{3}) - 1, 0), & \text{with probability } p_n(t), \\ v_n(t+\frac{2}{3}), & \text{default.} \end{cases}$$

Turn brake light on if appropriate:

$$\text{if } p = p_b \text{ and } v_n(t+1) < v_n(t+\frac{2}{3}), \quad \text{then } b_n(t+1) := on.$$

– Step 4: Move (drive):

$$x_n(t+1) := x_n(t) + v_n(t+1).$$

Free lane changes are needed so that vehicles can overtake slower driving passenger cars and trucks. When designing rules for the free lane changes, one should take care of that overtaking vehicles do not disturb the traffic on the lane they use to overtake to much, and one has to take account of German laws, which prohibit overtaking a vehicle to the left. Further, it is advantageous to prohibit trucks to drive on the leftmost lane in the simulation, because a truck overtaking another truck forces all vehicles on the left lane to reduce their velocity and produces a deadlock that may not resolve for a long time (14).

One more variable is needed for the free lane changes, $l_n \in \{left, right, straight\}$ notes if the vehicle $n$ should change the lane during the actual time-step or not. This variable is not needed if the lane changes are executed sequentially, but we prefer a parallel update. For the left free lane changes the simulator executes the following steps parallel for all vehicles $n$:

## *3.2. Overtake on the lane to the left:*

– Step 0: Initialization:

For car n find the next car m in front on the same lane, the next car *s* in front on the lane left to car *n*, and the next car *r* behind car *s*. Set $l_n$ := *straight*.

– Step 1: Check lane change:

if $b_n(t) = off$ and $v_n(t) > d_{n,m}(t)$ and $d_{n,s}^{eff}(t) \geq v_n(t)$ and $d_{r,n}(t) \geq v_r(t)$,
then set $l_n := left$.

– Step 2: Do lane change:

if $l_n =$ left, then let car *n* change lane to the left.

The definition of the gaps $d_{n,s}^{eff}(t)$ and $d_{r,n}^{eff}(t)$ in the lane-change-blocks is an obvious extension of the above definition; one simply inserts a copy of the car *n* on its left or right side. These overtake rules used by the simulator can verbally be summed up as follows: first, a vehicle checks if it is hindered by the predecessor on its own lane. Then it has to take into account the gap to the successor and to the predecessor on the lane to the left. If the gaps allow a safe change the vehicle moves to the left lane. For the right free lane changes the simulator executes the following steps parallel for all vehicles n:

## *3.3. Return to a lane on the right:*

– Step 0: Initialization:

For car *n* find the next car *m* in front on the same lane, the next car *s* in front on the lane right to car *n*, and the next car *r* behind car *s*. Set $l_n$ := straight.

– Step 1: Check lane change:

if $b_n(t) = off$ and $t_{n,s}^h(t) > 3$ and $(t_{n,m}^h(t) > 6$ or $v_n(t) > d_{n,m}(t))$ and $d_{r,n}(t) > v_r(t)$,
then set $l_n :=$ right.

– Step 2: Change lane:

if $l_n$ = right, then let car $n$ change lane to the right.

Thus, a vehicle always returns to the right lane if there is no disadvantage in regard to its velocity and it does not hinder any other vehicle by doing so.

It should be noted, that it is not possible to first check for all lane changes to the left and to the right and then perform them all in parallel without doing collision detection and resolution. This would be necessary because there are autobahns with three lanes and more. To overcome this difficulty, the lane changes to the left, i.e., overtake, are given a higher priority than the lane changes to the right. For a systematic approach to multi-lane traffic, i.e., lane-changing rules, see, for example, (19). For a detailed discussion of the different models see (3, 9, 23) and the references therein.

## 4. Validation of the Model

A core requirement in the discussion of the simulation model is the detailed comparison with empirical data. Only if the model maps the real world sufficiently, it is capable to deal as the kernel of the online-simulation. Furthermore, one of the most puzzling points for any model is to reproduce significant empirical data on a macroscopic and microscopic level as well as the empirical observed coexistence of stable traffic states and, especially, the upstream propagation of wide moving jams through both free flow and synchronized traffic with constant velocity and without disturbing these states (13).

In analogy to the empirical setup, the simulation data are evaluated by a virtual loop detector, i.e., the number of cars passing a given link is measured as well as their velocity. This allows for the calculation of aggregated minute data of flow, speed and occupancy like for the empirical data.

The simulation run emulates a few hours of highway traffic, including the realistic variation of the number of cars that are fed into the system. Thereby, a large input rate leads to the emergence of synchronized flow, whereas at small rates small short-living jams evolve, as expected, in the vicinity of on-ramps, because of the local perturbations (8).

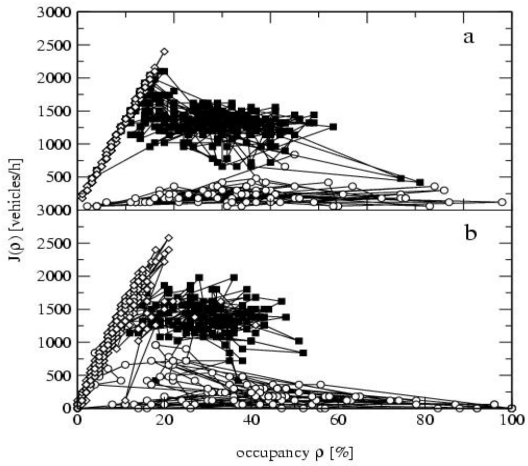


*Fig. 2: Comparison of the simulation results (a) with real traffic data (b). Diamonds correspond to free flow, squares to synchronized traffic, and circles to wide jams. Each point represents the average over an one-minute interval. The empirical data are from a detector on the A40 near Moers junction (synchronized state) and near Bochum-Werne.*

The simulation shows that the empirical results can quantitatively be recovered (see Fig. 2). A detailed analysis of the two dimensional region of synchronized traffic in the fundamental diagram reveals a high correlation between the two lanes with respect to the velocity time-series of both lanes.

## 5. Implementation of the Topology

An important point in the design of a simulator is the representation of the road network. Therefore, the network is divided into links. The main links connect the junctions and highway intersections representing the carriageway. Each junction and intersection consists of another link, like on/off-ramps or right/left-turn lanes. The attributes of each link are the length, the number of lanes, a possible speed limit, and the connecting links. In case of more than one connecting link, like at off-ramps or highway intersections, there is also a turning probability for each direction. The turning probability is calculated by taking into account the measured traffic data. All these spatial and functional data was collected to build a digital image of the topology of the whole network.

| | |
|---|---|
| Area | 34,000 km² |
| Inhabitants | 18,000,000 |
| On- and off-ramps | 862 |
| Intersections | 72 |
| Online loop detectors | 4,000 |
| Offline loop detectors | 200 |
| Number of links | 3,698 |
| Overall length | 2,250 km |

*Tab. 1: Design parameters of the North Rhine-Westphalian autobahn network.*

Another crucial information concerns the positions of the installed loop detectors. They also have to be included in the digital map of the network. The positions in the simulation are called ‘checkpoints’, and at these checkpoints the simulation is adapted to the measured traffic flow of the loop detectors. Tab. 1 shows some design parameters of the network. North Rhine-Westphalia is approximately one fifth of whole of Germany with respect to many numbers, e.g., number of cars, inhabitants, length of the autobahn network, et cetera.

## 6. Additional Rules for Complex Real Networks

The cellular automaton model for traffic flow used by the simulator was designed to be able to reproduce the main aspects of the fundamental diagram for real traffic flows (vehicle flow as a function of vehicles per km) and the fundamental microscopic properties, like the time headway distribution. This ability was verified by testing it on topologically simple networks. When simulating the traffic on a large and topologically complex network, like the autobahn network in North Rhine-Westphalia, some extensions to the cellular automaton model have to be considered. One is the guidance of vehicles and another is a strategy to integrate the measured flow from the loop detectors into the simulation.

A real driver usually has the intention to reach some goal with his driving. This makes it necessary to incorporate routes in the modeling. In principle, there are two different strategies to solve this problem. One can assign an origin and a destination to the road user and then guide him through the network according to this route (20, 22). For our network origin-destination information with a sufficient temporal and spatial resolution is not available. Therefore, the vehicles are guided in the network according to the probabilities calculated on the basis of the measured data. This means that a vehicle is not guided through the whole network, but every time it reaches a new link it will decide in accordance with the measured probabilities how it leaves the link.

To implement this we use forced lane changes. Forced lane changes are necessary so that the vehicles can drive from on-ramps on the autobahn, from the autobahn on off-ramps, when the autobahn narrows, and when vehicles drive from one particular section of the autobahn on another over an intersection. Forced lane changes differ from free lane changes in a fundamental way. While free lane changes give vehicles the opportunity to overtake vehicles driving slower and thus reduce disturbances, forced lane changes stem from the need to reach a node and are obviously an additional source for disturbances.

The simulator uses gradually increasing harsh measures to force lane changes. At the beginning of an area where a vehicle could change to the target lane, it does so, if the gap is sufficiently large and no vehicle is severely hindered. At the end of the area it will bully into any gap regardless of velocity differences. Further, a vehicle driving on its target lane should not leave the lane to overtake. An efficient implementation of this strategy is to store the lane change information in the cells. This gives a fast access through the coordinates of a vehicle. Of course this information depends on the node chosen and whether the vehicle is a truck or a passenger car. Because of this, every link has several versions of the lane change information.

To incorporate the real world measurements from the loop detectors into the simulation vehicle-moving, inserting, and removing algorithms have to be applied. This is done at the so-called checkpoints, which are located at those places in the network where a complete cross-section is available, i.e., all lanes are covered by a loop detector. Every time, when checkpoint-data is provided, the simulator uses the measured values to adjust the traffic state in the simulation. The first step is to try to move vehicles behind the checkpoint in front of it and vice versa. If this is not sufficient, vehicles are inserted or removed. This should be preferred to pure insert/removal strategies, as these can completely fail due to positive feedback if a non-existing traffic jam is produced by the simulation. In this case the simulation measures a low flow in comparison with the real data, so vehicles are added periodically to the ever growing traffic jam leading to a total breakdown.

## 7. The Website *www.autobahn.nrw.de*

The design of the simulator was financially supported by the Ministry of Transport, Energy and Spatial Planning of North Rhine-Westphalia, the reason being, that it wanted a novel web-based traffic information system for the public. This information system is provided by a Java applet at the URL *www.autobahn.nrw.de* (Fig. 3). The Java applet draws a map of North Rhine-Westphalia, where the autobahns are colored according to the level of service of the simulated traffic state, from light green for free flow, over dark green and yellow for dense and very dense synchronized flow, to red for a traffic jam. Additionally, after numerous requests, we integrated a color-blind mode, where dark green is replaced by dark grey and yellow by blue. Further, construction areas are drawn at the appropriate positions on the map and their estimated influence on the traffic is shown through red construction signs for a high risk of a traffic jam and green construction signs for a low risk. Road closures, which have a deep impact not only on the specific track the closure happens, but also on the traffic in a wide part of the network, are shown as well.

To make orientation easier the number and name of each junction is also written in the status bar when the mouse moves over the pictogram of the junction. All this valuable information assists the road user to choose the best route and the best time for his trip.

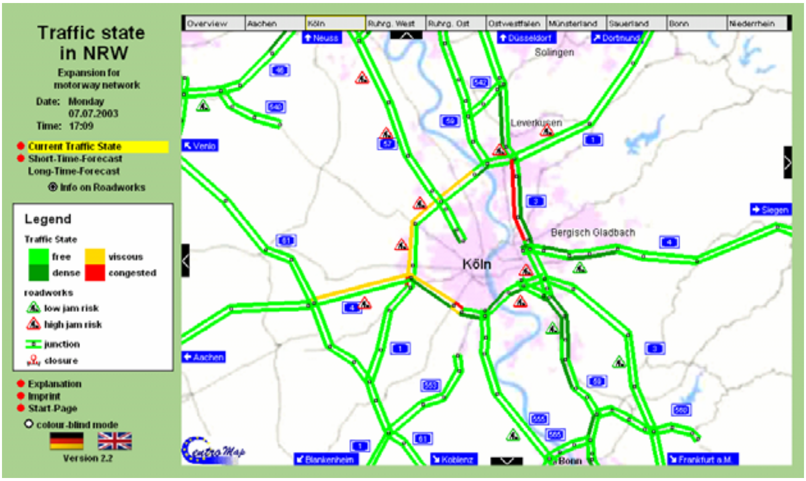


*Fig. 3: The current traffic state is visualized at www.autobahn.nrw.de using a Java applet.*

The rising accesses to OLSIM and the nearly throughout positive feedback show, that this information system is accepted by many people and used regularly. The daily requests increased from about 20,000 on work days at the beginning in September 2002 up to 200,000 regular accesses after the implementations of the 30 minute forecast in March 2003 and the 60 minute forecast in December 2003. The positive results are underlined by TV-stations, newspapers, and magazines which have made positive tests where they compared the actual traffic state to the traffic state presented by our simulation.

## 8. Summary

In this paper we present a new advanced traffic information system OLSIM which gives the internet user the opportunity to get the information about the current traffic state and a 30 and 60 minute prognosis of the autobahn network of North Rhine-Westphalia. The system rests upon a microscopic traffic simulator of the autobahn network in North Rhine-Westphalia. The simulator uses an advanced cellular automaton model of traffic flow and adjusts the traffic state in accordance with measurements of the real traffic flow provided by 4,000 loop detectors installed locally on the autobahn. The cellular automaton model, the abstraction of the network, the guidance of the vehicles, and the data integration strategies to periodically adjust the traffic flow in the simulation in accordance with the measured flow on the autobahn were discussed, as well as some details on the efficient implementation of the dynamics and the presentation of the simulated traffic state to the public. A graphical user interface implemented by a Java applet can be accessed by every internet user. In a simple to navigate window the user can choose between the current traffic state, the 30, and the 60 minute prognosis. Additional information like road works can be chosen with a simple click.

## References


(1) R. Barlovic, L. Santen, A. Schadschneider, M. Schreckenberg. “Metastable states in cellular automata for traffic flow.” Eur. Phys. J. B 5, 1998, pp. 793-800.

(2) C. Barrett, M. Wolinsky, M. Olesen. “Emergent local control properties in particle hopping traffic simulations.” In: (9), 2000, pp. 169-173.

(3) D. Chowdhury, L. Santen, A. Schadschneider “Statistical Physics of Vehicular Traffic and Some Related Systems”. Physics Reports 329, 2000, pp. 199-329.

(4) L.C. Edie, R.S. Foot. “Traffic flow in tunnels.” Proc. HRB 37, 1958, pp. 334-344.

(5) J. Esser, M. Schreckenberg. "Microscopic simulation of urban traffic based on cellular automata." Int. J. of Mod. Phys. C 8, 1997, pp. 1025-1036.
(6) H. George. "Measurement and Evaluation of Traffic Congestion." Bureau of Highway Traffic, Yale University, 1961, pp. 43-68.
(7) D. Helbing. "Empirical traffic data and their implications for traffic modelling". Phys. Rev. E 55, 1996, pp. R25-R28.
(8) D. Helbing, A. Henneke, M. Treiber. "Phase diagram of traffic states in the presence of inhomogenities." Phys. Rev. Lett. 8, 1999, pp. 4360–4363.
(9) D. Helbing, H. Herrmann, M. Schreckenberg, D.E. Wolf (Eds.) *Traffic and Granular Flow '99.* Springer, Heidelberg. 2000.
(10) Highway Capacity Manual. HRB Spec. Rep. 87. U.S. Department of Commerce, Bureau of Public Road, Washington, D.C., 1965.
(11) O. Kaumann, K. Froese, R. Chrobok, J. Wahle, L. Neubert, M. Schreckenberg. "On-line simulation of the freeway network of North Rhine-Westphalia." In: (9), pp. 351-356.
(12) B. Kerner, H. Rehborn. "Experimental properties of phase transitions in traffic flow." Phys. Rev. Lett. 79. 1997. pp. 4030-4033.
(13) B.S. Kerner. "Complexity of synchronized flow and related problems for basic assumptions of traffic flow theories." Network and Spatial Economics 1, 2001, pp. 35-76.
(14) W. Knospe, L. Santen, A. Schadschneider, M. Schreckenberg. "Disorder effects in cellular automata for two-lane traffic." Physica A 265, 1999, pp. 614-633.
(15) W. Knospe, L. Santen, A. Schadschneider, M. Schreckenberg. "Towards a realistic microscopic description of highway traffic." J. Phys. A 33, 2000, pp. L1-L6.
(16) W. Knospe. "Synchronized traffic: Microscopic modeling and empirical observations." Ph.D. Thesis, Gerhard-Mercator-University Duisburg, Germany, 2002.
(17) A. Miller. "A queuing model for road traffic flow." J. of the Royal Stat. Soc. B1, 23, University Tech. Rep. PB 246, Columbus, USA, 1961, pp. 69-75.
(18) K. Nagel, M. Schreckenberg. "A cellular automaton model for freeway traffic." J. Physique I 2, 1992, pp. 2221-2229.
(19) K. Nagel, D.E. Wolf, P. Wagner, P. Simon. "Two-lane traffic rules for cellular automata: A systematic approach." Phys. Rev. E 58, 1998, pp. 1425-1437.
(20) K. Nagel, J. Esser, M. Rickert. "Large-scale traffic simulations for transport planning." In: D. Stauffer (Ed.), Ann. Rev. of Comp. Phys. VII, World Scientific, Singapore, 2000, pp. 151-202.
(21) R.C. Pfefer. "New safety and service guides for sight distances." Transportation Engineering Journal of American Society of Civil Engineers 102, 1976, pp. 683-697.
(22) M. Rickert, P. Wagner. "Parallel real-time implementation of large-scale, route-plan-driven traffic simulation." Int. J. of Mod. Phys. C 7, 1996, pp. 133-153.
(23) M. Schreckenberg, D.E. Wolf (Eds.) *Traffic and Granular Flow '97.* Springer, Singapore, 1998.
(24) M. Schreckenberg, L. Neubert, J. Wahle. "Simulation of traffic in large road networks." Future Generation Computer Systems, 17, 2001, pp. 649-657.
(25) J. Treiterer. "Investigation of traffic dynamics by aerial photogrammatic techniques." Tech. report, Ohio State University Tech. Rep. PB 246, Columbus, USA, 1975.
(26) Q. Yang, H.N. Koutsopoulos. "A microscopic traffic simulator for evaluation of dynamic traffic management systems." Transp. Res. C 4, 1996, pp. 113-129.