OLSIM: Inter-urban Traffic Information

Daniel Weber, Roland Chrobok, Sigurdur Hafstein, Florian Mazur, Andreas Pottmeier, and Michael Schreckenberg

University Duisburg-Essen
Lotharstr. 1, 47048 Duisburg, Germany
{weber,chrobok,hafstein,mazur,pottmeier, schreckenberg}@traffic.uni-duisburg.de
http://www.traffic.uni-duisburg.de

Abstract. Nowadays detailed and reliable information about current and future traffic states is a crucial requirement not only for each modern traffic control centre, but also for driver accessible traffic information applications. In this work the novel traffic information system OLSIM is presented that is based on three components. The first is a highly realistic traffic flow model with which all vehicles in a large scale network are simulated. The second component consists of efficient data processing and forecast algorithms that form heuristics from a large database that is fed every minute by traffic data of 4,000 inductive loop detectors across the road network. The third is a graphical user interface which can be accessed at www.autobahn.nrw.de. More than 200,000 users each day indicate the importance of such a system.

1 Introduction

The vehicular traffic has risen in a dramatic manner, particularly in densely populated regions. Whereas at first the autobahns could handle the traffic demand easily, nowadays, the existing autobahn network has reached its capacity limit. This is in particular true for the German state of North Rhine-Westphalia with its large urban areas in the Rhine-Ruhr region (Dortmund, Essen, Duisburg, Düsseldorf) and around Cologne. The daily occurring traffic jams cause significant economic and ecological damage, and an enlargemant of the existing network is usually not possible. 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 (**O**n**L**ine Traffic **SIM**ulation) 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 traffic in North Rhine-Westphalia. Our approach to generate the traffic state in the whole autobahn network is to use locally measured traffic data, as the input into an advanced cellular automaton traffic simulator. These measured data, which are delivered minute by minute, are mainly provided by about 4,000 loop detectors and include especially the number of vehicles and trucks passed, the average

T. Böhme et al. (Eds.): IICS 2004, LNCS 3473, pp. 296–306, 2006.

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 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 Outline 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.

First of all the different kinds of data have to be collected. Especially, the traffic data are 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 a 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 minute, and for the 60 minute 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 section.

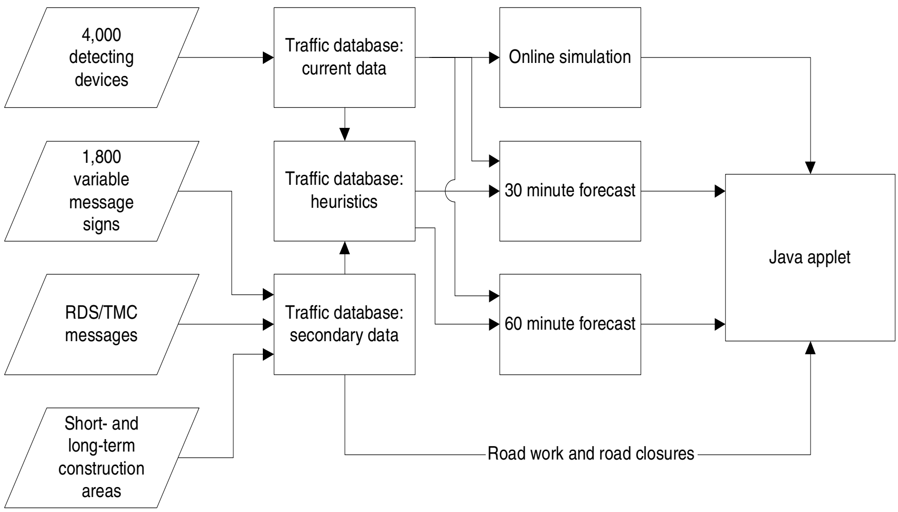

Fig. 1. The architecture of the traffic information system OLSIM

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 [1]. 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 in 1992 [2]. 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 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. 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. A vehicle occupies $2 - 5$ consecutive cells. By using velocity dependent randomization [3], 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 [4]. The inclusion of anticipation and brake lights [5] 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.

There is only one global parameter in the Nagel-Schreckenberg model, 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 $v_n(t) \in \{0, 1, ..., v_{max}\}$ at time t. When the vehicle n decides in the time-step $t \mapsto t + 1$ 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 few 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

$$d^{\text{eff}}_{n,m}(t) := d_{n,m}(t) + \max(v^{\min}_m(t) - d_S, 0)$$

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

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

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 of it. Brake lights are further components of the anticipated driving. They allow drivers to react to disturbances in front of them earlier by adjusting their speed. The variable $b_n(t) = \text{on}$ if car n has its brake lights on and $b_n(t) = \text{off}$ if they are off.

Several empirical observations suggest that drivers react in a temporal- rather than a spatial-horizon [7,8]. For this reason the velocity-dependent temporal interaction horizon

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

is introduced to 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 $t^S_n(t)$, i.e.,

$$t^h_{n,m}(t) := \frac{d_{n,m}(t)}{v_n(t)} < t^S_n(t) \ .$$

The estimations for h vary from 6 s [7], 8 s [8], 9 s [9] to 11 s [10]. Another estimation can be obtained from the analysis of the perception sight distance. In [11] 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 [12].

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

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

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 [12]. To sum up, to move the vehicles forward in the network the algorithm executes the following steps in parallel for all vehicles n:
Move forward (drive):

- Step 0: Initialization:
  For car n find the next car m in front.
  Set $p_n := p(v_n(t), b_m(t))$ and $b_n(t+1) :=$ off.

- Step 1: Acceleration:

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

- Step 2: Braking:

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

  Turn brake light on if appropriate:

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

- Step 3: Randomization with probability p_n:

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

Turn brake light on if appropriate:

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

- 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.

One more variable is needed for the free lane changes, $l_n \in \{\text{left}, \text{right}, \text{straight}\}$ 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 of the lane changes for all vehicles and that renders this variable necessary. For the left free lane changes the simulator executes the following steps parallel for all vehicles n:
Overtake on the lane to the left:

- Step 0: Initialization:
  For car n find the next car m in front on the same lane, 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:

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

- Step 2: Do lane change:

$$\text{if } l_n = \text{left, then let car } n \text{ change lane to the left.}$$

The definition of the gaps $d^{\text{eff}}_{n,s}(t)$ and $d_{r,n}(t)$ in the lane-change-blocks is an obvious extensions 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:
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 $t^h_{n,s}(t) > 3$ and $(t^h_{n,m}(t) > 6$
  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 if 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, [13]. For a detailed discussion of the different models see, e.g., [6] and the references therein.

4 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 consist 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.

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. Table 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.

Table 1. Design parameters of the North Rhine-Westphalian autobahn network

Area	34,000 km^2
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

5 Traffic Data and Forecast

The simulation describes the dynamics in a network but lacks information about the boundaries. Especially for forecasts, reasonable data has to be incorporated. Many approaches to predict traffic states have been investigated in the past, such as Box Jenkins techniques [14], heuristics [15,16], neural networks [17,18], Kalman filtering [19], and nonparametric regression [20], as well as several comparisons [21] and combinations [22]. The result of the forecast methods strongly depend on the prognosis horizon. For short-term forecasts the current traffic data is of an enormous importance and many complex forecast methods are outperformed by simple models like the single smoothing average or the moving average model. For longer horizons the experience from the past in form of heuristics creates better results.

Because of this fact different forecast methods for the 30 and the 60 min traffic predictions are used. After an intensive statistical analysis of different kinds of traffic data [23], a 14-day classification is chosen considering the different day to day traffic. The average flow and velocity data $x_{\mathrm{hist}}(t_{\mathrm{p}})$ of the last 20 days of each class is averaged and used as a pattern for long-term forecasts. As the short-term forecast method the smoothing averaged values of the last recent minutes $x_{\mathrm{c}}(t_0)$ are used.

Finally, for the prognosis horizons $\Delta\tau = 30$ min and $\Delta\tau = 60$ min the predicted traffic value $x_{\mathrm{pred}}(t_{\mathrm{p}})$ at the time t_p is the sum of the average $x_{\mathrm{hist}}(t_{\mathrm{p}})$ of the sample class and the difference of $x_{\mathrm{c}}(t_0)$ and $x_{\mathrm{hist}}(t_{\mathrm{p}})$ weighted with k:

$$x_{\mathrm{pred}}(t_{\mathrm{p}}) = x_{\mathrm{hist}}(t_{\mathrm{p}}) + k \cdot \Delta x(t_0),$$

with

$$\begin{aligned} \Delta x(t_0) &= x_{\mathrm{c}}(t_0) - x_{\mathrm{hist}}(t_0), \\ k &= \begin{cases} \eta\left(1 - \frac{\Delta\tau}{\Delta\tau_{\max}}\right), & \text{if } 0 < \Delta\tau \leq \Delta\tau_{\max} \\ 0, & \text{if } \Delta\tau > \Delta\tau_{\max} \end{cases} \\ \Delta\tau &= t_{\mathrm{p}} - t_0. \end{aligned}$$

t_0 is the point in time when the forecast is made. Obviously, for $\Delta\tau > \Delta\tau_{\max}$ the heuristics is used as forecast. The factor η is a coefficient determining the relevance of the current and historical data respectively. Reasonable is $0 < \eta < 1$; for $\eta = 0$ only the heuristics are used, for $\eta = 1$ the prognosis uses the current value.

6 Implementation of Traffic Data

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 enough to adjust the traffic state, vehicles are inserted or removed. This should be preferred to pure insert/removal strategies, because 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 Web-Based Information

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 `www.autobahn.nrw.de` (see Fig. 2). 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 starting time for his trip.

The rising accesses to OLSIM and the nearly throughout positive feedback shows 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.

Up to now, we have restricted OLSIM on the autobahn network. The next step will be to incorporate the national streets (*Bundesstrassen*) as well as some of the heavily used urban streets.

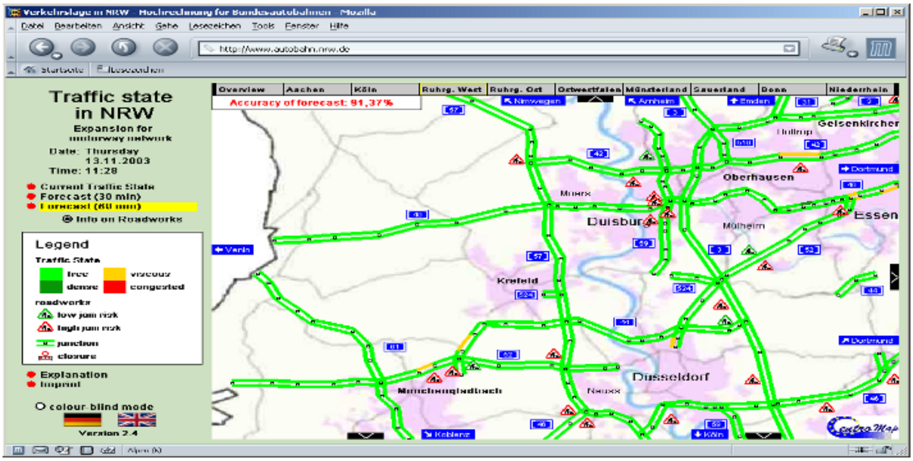

Fig. 2. Screen shot of the visualization of the 60 minute traffic forecast of North Rhine-Westphalia (here one can see the area around Duisburg). For further information about this application and to see the current and future traffic state see: `http://www.autobahn.nrw.de`

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 which 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. J. Wahle, R. Chrobok, A. Pottmeier, and M. Schreckenberg.: A Microscopic Simulator for Freeway Traffic. Network and Spatial Economics **2** pp. 371–386, (2002).
2. K. Nagel and M. Schreckenberg.: A cellular automaton model for freeway traffic. J. Physique I **2**, pp. 2221–2229, (1992).

3. R. Barlovic, L. Santen, A. Schadschneider, and M. Schreckenberg.: Metastable states in cellular automata for traffic flow. Eur. Phys. J. B **5**, pp. 793–800, (1998).
4. D. Helbing.: Empirical traffic data and their implications for traffic modelling. Phys. Rev. E **55**, pp. R25–R28, (1996).
5. S.F. Hafstein, R. Chrobok, A. Pottmeier, J. Wahle, and M. Schreckenberg.: Cellular Automaton Modeling of the Autobahn Traffic in North Rhine-Westphalia. In: I. Troch and F. Breitenecker (Eds.), Proc. of the 4th MATHMOD, Vienna, Austria, pp. 1322–1331, (2003).
6. D. Helbing, H.J. Herrmann, M. Schreckenberg, and D.E. Wolf (Eds.).: Traffic and Granular Flow '99, Springer, Heidelberg, (2000).
7. H.P. George.: Measurement and Evaluation of Traffic Congestion. Bureau of Highway Traffic, Yale University, pp. 43–68, (1961).
8. A. Miller.: A queuing model for road traffic flow. J. of the Royal Stat. Soc. B1, 23, University Tech. Rep. PB 246, Columbus, USA, pp. 69–75, (1961).
9. Highway Capacity Manual. HRB Spec. Rep. 87. U.S. Department of Commerce, Bureau of Public Road, Washington, D.C., (1965).
10. L.C. Edie and R.S. Foot.: Traffic flow in tunnels. Proc. HRB **37**, pp. 334–344, (1958).
11. R.C. Pfefer.: New safety and service guides for sight distances. Transportation Engineering Journal of American Society of Civil Engineers **102**, pp. 683–697, (1976).
12. W. Knospe.: Synchronized traffic: Microscopic modeling and empirical observations. Ph.D. Thesis, University Duisburg-Essen, Germany, (2002).
13. K. Nagel, D.E. Wolf, P. Wagner, and P. Simon.: Two-lane traffic rules for cellular automata: A systematic approach. Phys. Rev. E **58**, pp. 1425–1437, (1998).
14. S.A. Ahmed.: Stochastic Processes in Freeway Traffic. Transp. Res. Record **19–21**, pp. 306–310, (1983).
15. D. Wild.: Short-term forecasting based on a transformation and classification of traffic volume time series. Int. J. of Forecasting **13**, pp. 63–72, (1997).
16. S. van Iseghem and M. Danech-Pajouh.: Forecasting Traffic One or Two Days in Advance - An Intermodal Approach. Recherche Transports Securite **65**, pp. 79–97, (1999).
17. M.S. Dougherty and M.R. Cobbett.: Short-term inter-urban traffic forecasts using neural networks. Int. J. of Forecasting **13**, pp. 21–31, (1997).
18. H. Dia.: An object-oriented Neural Network Approach to Short-Term traffic forecasting. Euro. J. Op. Res. **131**, pp. 253–261, (2001).
19. J. Whittaker, S. Garside, and K. Lindveld.: Tracking and Predicting a Network Traffic Process. Int. J. of Forecasting **13**, pp. 51–61, (1997).
20. A.S. Nair, J.-C. Liu, L. Rilett, and S. Gupta.: Non-Linear Analysis of Traffic Flow. In: B. Stone, P. Conroy, and A. Broggi (Eds.), Proc. of the 4th International IEEE Conference on Intelligent Transportation Systems, Oakland, USA, pp. 683–687, (2001).
21. H.R. Kirby, S.M. Watson, and M.S. Dougherty.: Should we use neural networks or statistical models for short-term mororway traffic forecasting? Int. J. of Forecasting **13**, pp. 43–50, (1997).
22. M. Danech-Pajouh and M. Aron.: ATHENA: a method for short-term inter-urban motorway traffic forecasting. Recherche Transports Sécurité - English issue **6**, pp. 11–16, (1991).
23. R. Chrobok, O. Kaumann, J. Wahle, and M. Schreckenberg.: Different methods of traffic forecast based on real data. Euro. J. Op. Res. **155**, pp. 558–568, (2004).