ABSTRACT
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 named OLSIM is based on an efficient and highly realistic traffic flow model, which is fed by traffic data from 4,000 detecting devices across the road network every minute, and a graphical user interface which can be accessed at http://www.autobahn.nrw.de.

KEYWORDS
Traffic Information, Traffic Forecast, Online-Simulation (OLSIM), Real Time Systems, Internet Applications

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 on. New information systems and traffic management concepts are thus truly needed.

Therefore, we established the new advanced traffic information system OLSIM which gives the internet user the opportunity to get information about the current traffic state, a 30 minute and a 1 hour prognosis of the traffic on 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 and infrared or radar detection devices, 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 delivers 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 give a short-term forecast for 30 minutes, which was extended in the next step to 1 hour. These information are completed by the temporal and spatial road work and actual road closures. All these valuable traffic information are integrated in a Java-Applet that can be accessed by every internet user at http://www.autobahn.nrw.de.
In this paper we will briefly discuss the general concept of the traffic information system OLSIM. The main focus lies then on the core module of the system, the online traffic simulator. We give a detailed description of the microscopic model we use as the kernel of the simulation, the complex topology of the road network, and some challenges encountered when using this model on such a huge network. We finish with a short look at the web site http://www.autobahn.nrw.de. Beside a simple user interface for the current traffic, the 30 minutes and the 1 hour prognosis, it provides additional information like temporal and spatial information of road works and road closures.

2. GENERAL CONCEPT OF THE TRAFFIC INFORMATION SYSTEM

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 Figure 1.

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 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 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 by 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 by the Java-Applet at the website http://www.autobahn.nrw.de. Traffic state, construction areas, and road closures are integrated into 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 minutes prognosis.

The microscopic traffic simulation, on which the core of the information system is based, is focused on in the next sections. 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

Numerous approaches have been proposed for modeling vehicular traffic in the past. The approaches are usually classified into microscopic- and macroscopic models by the way the vehicle movements are considered. The macroscopic models are restricted to collective vehicle dynamics and describe these, in analogy to the hydrodynamical description of compressible viscous fluids, in terms of macroscopic quantities like flow and density. The most widespread approach describes the time-evolution of the density and the flow by the continuity-equation (conservation of vehicles) [Kerner, 1994, Lee, 1998]. Other macroscopic models are based on the gas-kinetic approach, whereby the macroscopic model is derived from microscopic equations. In the gas-kinetic approach one considers the probability density of the number of vehicles as function of time, position and velocity and the time-evolution is described by a Boltzmann-like equation [Prigogine, 1960]. A detailed discussion of the macroscopic models would extend the scope of this paper, the interested reader is referred to the literature.

The microscopic models focus on the reproduction of the movements of the individual vehicles. Thus, important quantities like travel times, routes, velocity profiles and lane-change behavior are directly accessible in the microscopic models, in contrast to macroscopic models, where these quantities have to be derived using some additional assumptions. In this paper we focus on cellular automaton modeling of vehicular traffic. Other significant approaches are, for example, the car-following models [Chandler, 1958, Reuschel, 1950], the optimal-velocity models [Bando, 1998] and the intelligent driver model [Treiber, 2000].

The kernel of the online simulation is an advanced and highly realistic microscopic 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 [Esser, 1997, Kaumann, 2000, Schreckenberg, 2001, Nagel, 2000, Rickert, 1996]. 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 [Nagel, 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/s² (0-100 km/h in 19 s) which is of the same order of the “comfortable” acceleration of about 1 m/s². By using velocity dependent randomization [Barlovic, 1998], realized through the introduction of ‘slow-to-start rules’, metastable traffic flows can be reproduced in the simulation, a phenomenon observed in empirical studies of real traffic data [Helbing, 1996, Treiterer, 1975, Kerner, 1997]. The inclusion of anticipation and brake lights [Barrett, 2000, Knospe, 2000] 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 $v_n(t) \in \{0, 1, \ldots, v_{\text{max}}\}$ at time $t$. When the vehicle $n$ decides in the time-step $t \rightarrow t+1$ 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

\[
d_{\text{eff}}(t) := d_{mn}(t) + \max(v_{mn}(t) - d_s, 0)
\]

seen by vehicle \( n \) at time \( t \). In this formula \( d_s \) is a safety distance and

\[
v_{mn}^{\text{max}}(t) := \min(d_{mn}(t), v_n(t)) - 1,
\]

is a lower bound of how far the vehicle \( m \) will move during this time-step. \( d_{mn}(t) \) is the number of free cells between car \( m \) and car \( l \) in front of it. Brake lights are further components of the anticipating driving. They allow drivers to react to disturbances in front of them earlier by adjusting their speed. The variable \( b_m(t) = \text{on} \) if car \( n \) has its brake lights on and \( b_m(t) = \text{off} \) if they are off. Several empirical observations suggest that drivers react in a temporal- rather than a spatial-horizon [George, 1961, Miller, 1961]. For this reason the velocity-dependent temporal interaction horizon

\[
t^*_{mn}(t) := \min(v_n(t), h)
\]

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 \( v_n = \text{const.} \) and car \( m \) standing still, is less than

\[
t^*_{mn}(t) := \frac{d_{mn}(t)}{v_n(t)} < t^*_{mn}(t).
\]

The estimations for \( h \) vary from 6 s [George, 1961], 8 s [Miller, 1961], 9 s [Highway Capacity Manual, 1965] to 11 s [Edie, 1958]. Another estimation can be obtained from the analysis of the perception sight distance. In [Pfeifer, 1976] 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 [Knospe, 2002].

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

\[
p = p(v_n(t), b_m(t)) := \begin{cases} p_s, & \text{if } b_m(t) = \text{on} \text{ and } t^*_{mn}(t) < t^*_{mn}(t), \\
p_0, & \text{if } v_n(t) = 0, \\
p_d, & \text{default.}
\end{cases}
\]

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. Leaving \( h = 6 \) fixed, the best agreement can be achieved for \( d_s = 7 \) cells, \( p_0 = 0.96, p_0 = 0.5, \) and \( p_d = 0.1. \) A detailed comparison of the fundamental diagrams, the time headway distributions, the lane usage, and the autocorrelations and crosscorrelations of density, velocity, and flow in the model presented here with empirical data and earlier models would significantly extend the scope of this contribution. For a thorough discussion on these and the calibration of the parameter-set we also refer to [Knospe, 2002].

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 \( p_d(t) := p(v_n(t), b_m(t)) \) and \( b_d(t+1) := \text{off}. \)

- **Step 1: Acceleration:**
  \[
  v_n(t + \frac{1}{3}) := \begin{cases} v_n(t), & \text{if } b_m(t) = \text{on} \text{ or } (b_m(t) = \text{on} \text{ and } t^*_{mn}(t) < t^*_{mn}(t)), \\
  \min(v_n(t) + 1, v_{\text{max}}), & \text{default}.
\end{cases}
  \]

- **Step 2: Braking:**
  \[
  v_n(t + \frac{2}{3}) := \min(v_n(t + 1/3), d_{\text{eff}}(t)).
  \]
Turn brake light on if appropriate:
\[ \text{if } v_r(t + \frac{2}{3}) < v_s(t), \text{ then } b_s(t+1) := \text{on}. \]

- Step 3: Randomization with probability \( p_n(t) \):
\[ v_s(t+1) = \begin{cases} \max(v_s(t + \frac{2}{3}) - 1, 0), & \text{with probability } p_n(t), \\ v_s(t + \frac{2}{3}), & \text{default.} \end{cases} \]

Turn brake light on if appropriate:
\[ \text{if } p = p_n \text{ and } v_r(t + 1) < v_s(t + \frac{2}{3}), \text{ then } b_s(t+1) := \text{on}. \]

- 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 [Knospe, 1999]. One more variable is needed for the free lane changes, \( l_0 \in \{ \text{left, right, straight} \} \) 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 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 \( n \).

### 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 \( l_0 := \text{straight} \).
  
  - Step 1: Check lane change:
    \[ \text{if } b_r(t) = \text{off} \text{ and } v_s(t) > d_{n,m}(t) \text{ and } d_{n,s}^{\text{off}}(t) \geq v_s(t) \text{ and } d_{n,r}(t) \geq v_s(t), \text{ then set } l_0 := \text{left}. \]
  
  - Step 2: Do lane change:
    \[ x_n(t + 1) = x_n(t) + v_n(t + 1). \]

The definition of the gaps and 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 \( l_0 := \text{straight} \).
  
  - Step 1: Check lane change:
    \[ \text{if } b_{r}(t) = \text{off} \text{ and } v_s(t) > d_{n,m}(t) \text{ and } d_{n,s}^{\text{off}}(t) \geq v_s(t) \text{ and } d_{n,r}(t) > v_s(t), \text{ then set } l_0 := \text{right}. \]
  
  - Step 2: Change lane:
    \[ x_n(t + 1) = x_n(t) + v_n(t + 1). \]
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, [Nagel, 1998]. For a detailed discussion of the different models see [Chowdhury, 2000, Helbing, 2000, Schreckenberg, 1998] 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 (Figure 2). 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. There are two types of detectors: online and offline. The online detectors are directly connected to the simulation and provide traffic data every minute. Since the offline detectors are not accessible by our computer network, only a sample data set can be used. 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

<table>
<thead>
<tr>
<th>Area</th>
<th>34,000 km²</th>
<th>Inhabitants</th>
<th>18,000,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>On- and off-ramps</td>
<td>862</td>
<td>Number of links</td>
<td>3,698</td>
</tr>
<tr>
<td>Intersections</td>
<td>72</td>
<td>Overall length</td>
<td>2,250 km</td>
</tr>
<tr>
<td>Online loop detectors</td>
<td>4,000</td>
<td>Offline loop detectors</td>
<td>200</td>
</tr>
</tbody>
</table>
5. VISUALISATION OF THE TRAFFIC STATE

The design of the simulator was financially supported by the Ministry of Transport, Energy and Spatial Planning of the state of North Rhine-Westphalia. The reason is, 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 http://www.autobahn.nrw.de. The Java-Applet draws a map of NRW, where the autobahns are coloured according to the simulated traffic state, from light green for free flow, over dark green and orange yellow for dense and very dense synchronised flow, to red for a traffic jam. Additionally, after numerous requests, we integrated a colour-blind mode where dark green is replaced by dark grey and orange yellow by blue. 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 are drawn too and by moving the mouse cursor over a construction area or a road closure the appropriate notification from the traffic centers or the police are shown in a tooltip. Further, a 30 minutes and a 1 hour traffic forecast are integrated in the application (Figure 3).

![Figure 3. The current traffic state, the 30 minutes and the 1 hour forecast](image)

The resonance to this novel information system has been very positive and TV-stations, newspapers, and magazines have made positive tests where they compare the actual traffic state to the traffic state and traffic forecast presented by our simulation. However, the crucial acknowledgement for the utility of our web-based traffic information service are the 250,000 users every day.

6. CONCLUSION

In this paper we present the new advanced traffic information system OLSIM, which gives the internet user the opportunity to get the information about the current traffic state, a 30 minutes and a 1 hour prognosis for 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 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 and the prognoses.
REFERENCES