Traffic simulation using agent-based models

Conference Paper · October 2009

DOI: 10.1109/ICAT.2009.5348417

Vedran Ljubovic
University of Sarajevo

Traffic Simulation Using Agent-based Models

Vedran Ljubović
vedran.ljubovic@etf.unsa.ba
Elektrotehnički fakultet Sarajevo
Zmaja od Bosne b.b.
71000 Sarajevo, BiH

***Abstract*-In this paper we will build a computer traffic model simulating movement of each individual vehicle through the traffic network and the interactions of that vehicle with other vehicles and semaphores (an agent-based model). It will model a simple traffic network (a two-way coordinated semaphore system a.k.a. "green wave") to test certain hypotheses on different kinds of semaphore systems. We will show that, with certain limitations , such a model can run on an average PC computer at speed up to ten times the real-time. In order to validate this model, a two-way coordinated semaphore system will be statistically compared to a non-coordinated system, hopefully proving the advantage of semaphore coordination.**

I. Introduction to Agent-Based Models

Traditional approach to modeling usually assumes that behavior of each individual can be described formally with some simple equations. In the case of social models, this assumes rational and informed behavior of each individual. Based on these equations, a system of aggregate, usually differential equations, is formed to describe the model. Such model unfortunately doesn't reflect the reality of human systems as humans usually act irrationally, selfishly, they make mistakes in judgment. Besides, differences among individuals are large enough to have influence on the system, they can't simply be averaged out. Such problems are addressed by introducing various stohastic or calibration factors which are then compared to real, measured data.

Agent-based modeling (ABM) is a very powerful method for making simulations which is being used for multitude of real-life problems as of lately [1] [2]. ABM authors do not attempt to describe the whole system mathematically. Instead, the system is modeled as a collection of autonomous units that make decisions – agents. These agents make decisions based on their own perspective and decision rules, while lacking a complete picture of the system [1] [2]. It appears that from interactions of these relatively simple units, the same complex rules predicted by classical models emerge (emergent behavior) [1] [2].

In addition, from use of agent-based models it is often possible to infer rules and behaviors that couldn't have been predicted mathematically, even ones apparently counter to logic. A common example given for this in literature is the traffic jam which moves along the road in a direction opposite to the actual movement of vehicles [1].

ABM is best used in the following situations [2]:

1. When behavior of individual units is non-linear, or it can only be described by a combination of if-then rules and thresholds.
2. When individual unit behavior includes memory, path dependency / hysteresis, non-Markov behaviors or temporal dependencies such as adaptation and learning.
3. When interactions among agents are heterogeneous and can lead to network effects. With flow equations a homogeneous mixture is assumed, but due to the complex interaction topology such interactions can affect the whole system.
4. When averaging out eliminates important aspects of individual behavior that can affect the whole system.

An important advantage of ABM is that it enables a natural way of describing the system [3]. With traditional modeling approach, the modeler usually takes into account both of individual behavior and observed behavior of the whole system. ABM enables modeler to describe the individual unit using the language of procedural statements that is familiar to programmers, but is also easier to understand for domain experts who don't necessarily have formal education in mathematics.

Model of an individual agent can be separately calibrated and test. For example, in traffic models it is possible to measure behavior of individual drivers and compare it with the one encoded in agent model. After that, the agent is placed into a simulation so that the whole system behavior could be calibrated with measured values. Stohastic behavior aspects can be applied to the exact spot determined by the agent model as opposed to a generic "noise factor" used in system equations.

In this paper we will present the state of the art in the area of traffic modeling (Chapter II). We will also describe in detail our model developed using the ABM approach (Chapter III). Finally, this model will be tested on a specific example of prediction of behavior in a sequence of intersections with semaphore coordination (Chater IV). It will be shown that the observed model results match the predictions which further verifies the ABM approach. Chapter V gives a conclusion and outlines possible further work in this area.

This paper contributes to the body of knowledge in the researched area by validating and practically confirming the assumptions on ABM approach presented below. While we were unable to find papers that specifically present performance as a problem in traffic modeling, to us it is an obvious fact of life that doesn't need to be proven. We've also cited typical execution times for various modeling approaches, showing that ABM provides a good compromise between execution speed and precision.

II. Overview of Approaches to Traffic Modeling

Traditional approach to modeling traffic in city road networks is based on the concept of *equilibrium* where a fixed matrix of roads in the period evaluated is represented with a network where the time to traverse each individual link can be precisely defined with a monotonously increasing function of cost or flow.

Given the equilibrium is valid, choice of path or paths is guided by Wardrop principles where all chosen paths have equal and minimal traversal costs.[1] Thus, a very concrete assumption is made, not just on the number of drivers traveling each source-destination pair, but also which road they will take and what will the total travel time be. In addition, variations in traffic within the observed time period are generally ignored [5].

Obviously such picture is a considerable simplification of reality. Number of trips o each source-destination pair varies from day to day, and the drivers themselves behave differently. Road conditions also vary from day to day, in part due to fluctuations in traffic, in part due to factors such as weather conditions, accidents etc. Finally, all drivers aren't equally successful at choosing the optimal path, especially the infrequent drivers. There is clear proof that these variations in "supply and demand" can affect the mean values in traffic such as travel time or fuel consumption [5]. For example, one paper found that the average travel time is 14% higher than equilibrium in the example of road network of northern Leeds (quoted from [5]).

Microsimulation or *microscopic traffic models*, unlike the above described approach, observes each individual vehicle in the traffic, and attempts to describe it with am ordinary differential equation. Such simulations are also referred to as *time-continuous simulations*.

A *car-following model* attempts to describe vehicle behavior based on the distance from the vehicle infront of it and the physical road characteristics. Most modern applications is

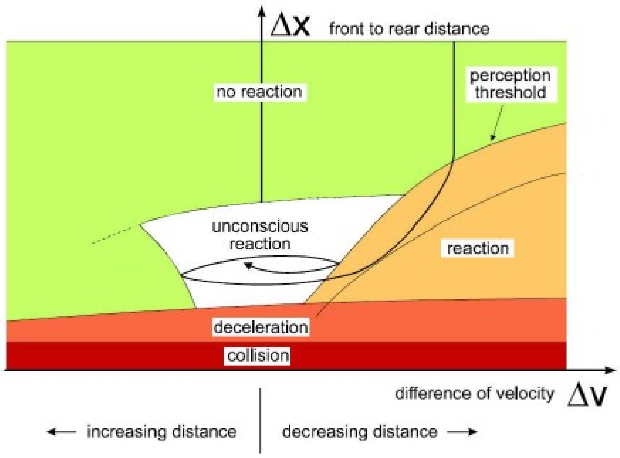

Figure 1. Wiedemanns car-following model [17]

based on Wiedemanns model for driver or vehicle behavior.[2] [6] This model assumes that each individual driver can be observed in one of the following states (see Fig. 1):

- *Free driving*, that is, an effect of another vehicle can not be noticed. In this regime a driver strives to reach and maintain a certain fixed speed, which is drivers desired speed. We can say that the speed asymptotically aproaches the ideal. In reality, free driving speed isn't perfectly constant but oscillates around the target speed due to imperfections in throttle control.
- *Approach*
- *Following*
- *Braking*

A feature of each mode is that the acceleration can be described as a function of current speed, difference in the speed of the vehicle in front, distance from the vehicle in front, and individual characteristics of driver and vehicle. A change in mode happens when certain thresholds are reached, again described with speed difference and distance. Small speed difference enables slower response and vice versa. The ability to assess speed and safety distance also varies among drivers. Therefore, this model is sometimes described as psycho-physical car following model.

A market leader in traffic simulations is a software named VISSIM which uses Wiedemanns model applied to discrete time units. VISSIM uses an improved version of Wiedemanns model with a number of calibration parameters.[6] In addition to car-following, VISSIM models pedestrian behavior using a social force model.

It is generally believed that macrosimulation approach is suitable for simulating traffic on highways, while microscopic simulation is necessary for precise analysis of traffic jams in urban situations [5]. Main shortcomings of microsimulation are the extremely long execution time, as well as high development costs due to lack of appropriate tools for

1 First Wardrop principle is as follows: "The time to traverse all chosen paths is equal or less than an individual driver would experience on any of the paths that weren't chosen". The Second principle is: "In the state of equilibrium the average travel time is minimal" [4].

2 This model is described in a paper that isn't available to us, but a sufficient description is given in [6].

supporting such modeling approach [7].

Models based on cellular automata have recently been proposed as an alternative to microscopic simulations based on continuous values. With cellular automata (CA) time and space are represented with integers. We can imagine that the space is divided into rectangular cells, each of which can contain a vehicle with certain characteristics. The main advantage of such approach is a very efficient implementation, which makes them suitable for simulating very large traffic networks [8].

A famous model of this type is Nagel-Schreckenberger (NaSch) model [9]. Its claim to fame is that this model was the first to successfully demonstrate "jam from nothing" phenomenon. Namely, many drivers have noted that on highways jams can sometimes appear for no particular reason: it's sufficient that one driver breaks unnecessarily and the line will be formed after him/her. Models based on differential equations failed to describe this phenomenon [10].

We will provide some details on the NaSch model, as given in [9] and [10], to illustrate both the strengths and the limitations of CA-based traffic models. Simulated space in NaSch model consists of square cells whose side is 7,5 meters (average car length + minimal gap between cars). One clock tick corresponds to one second, from which follows that car speed is expressed as integer multiple of 7,5 meters per second (cca. 27 km/h). Maximum speed is usually 5 cells per second (cca. 135 km/h). Model describes ideal driver behavior in which there is no overtaking, accidents etc.

Such a model is considered minimal and is useful for educational purposes in computer science, but in real use it can be complemented with further rules of driver behavior. Also, its simplicity makes it amenable to parallelization [11]. However, it must be observed that driver behavior is most simplified, not unlike macrosimulation approach.

A sample application of an improved NaSch model is OLSIM. Here, real-time ability of NaSch is used to present OLSIM users the current congestion of highways in German province Nordrhein-Westfalen, measured by 4000 traffic detectors, which is then used to predict the situation in 30 and 60 minutes. Prognosis are based on multiple heuristic schemes classified by days (workdays vs. weekend, holidays etc.) as well as road works, changing traffic signs and other special events. NaSch model used in OLSIM was improved considerably [8].

Agent-based traffic models have the potential of gaining advantages of both microscopic simulations based on mathematical models and models based on cellular automata. On one hand, an agent can include more detailed and realistic models of driver behavior; on the other, ABM doesn't require solving differential equations, which enables easier development. In addition, execution is faster then microsimulation which enables fine corrections and what-if analyses which so far where only available with CA-based models.

An important difference of ABM over CA is that with ABM only time is discreet, while space can be continuous. In each simulation round an agent is invoked to execute its actions (in our case – movement) [12]. As with microsimulation, ABM allows changing simulation time (tick). However, this can affect simulation results: shorter tick gives an agent more opportunities to evaluate the situation and thus gives a more nuanced response. Generally, by analogy to other discreet systems, it is assumed that results are a closer match to reality as the tick approaches zero.

There is a number of environments and frameworks that accelerate and simplify development of agent-based models, and faster development is further supported by a natural way of describing driver behavior [12] [7]. Also, using agents gives a better potential for parallelization [11] [7].

One trend in traffic ABM is using reactive models of driver behavior based on neural networks (intelligent agents). Such models closely match real drivers on the scale of individual drivers, but their use in modeling requires gathering data for training the neural network which proves to be a problem [13] [14].

The best known and widest used traffic ABM software is TRANSIMS, developed by Los Alamos National Laboratory for the needs of US Department of Transport. This software used to use NaSch cellular automata model improved with rules for lane-shifting, turning, signalization, parking etc. [15], however later versions are advertised as using ABM. We have no further information on improvements made since.

III. An Agent-Based Traffic Model - Description

Modeling requires above all to determine those features that are relevant to the model in question, and those features will be modeled while others will be neglected. In this paper, the most important parameters were *simplicity of code development*, *speed of execution*, while also considering our intended purpose which is *modeling traffic in a city*. Given that, our choice was to use agent-based modeling. Further consequence of this approach were a number of compromises:

1. Driver behavior is simplified to overtaking and intersection behavior, as we believe that reactive behavior isn't relevant given the scope (see [6]).
2. The car-following model is mostly based on Wiedemanns work (described in [6]) with several improvements regarding drivers' ability to determine speed of other cars, and the average inter-car distance.
3. Model assumes that drivers lack information on traffic congestion and therefore choose the shortest path with punitive factors for turning in intersections (waiting for right of way).
4. Model lacks support for internal traffic (parked vehicles), which isn't a problem for small segments of road network.
5. Varying vehicle lengths weren't considered. This is a possibility for future work, as further research showed

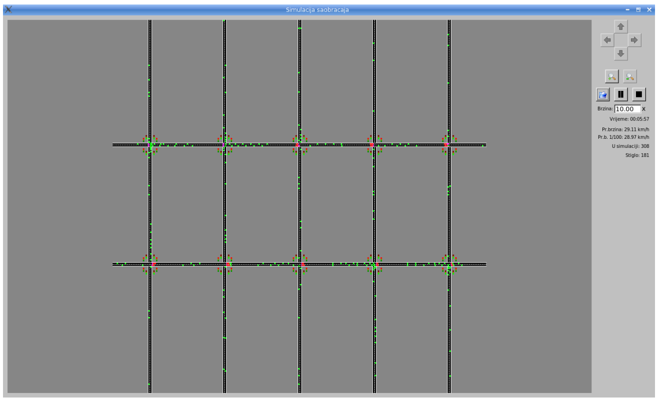

Figure 2. Screenshot showing simulation of two one-way streets

that such feature is easy to implement.

Of course, it is possible to create a model on different assumptions. Analysis of how those changed assumptions reflect on the modeling results is beyond the scope of this paper.

Given above assumptions, the model roughly works as follows:

1. A number of entrances and exits are defined, along with volumes of vehicles using given entrance/exit.
2. From these volumes, probabilities are calculated that in each moment a car will appear at given entrance. When a vehicle appears, its exit is also determined.
3. For each car optimal path is determined.
4. Each car also receives its ideal speed.
5. After that, agent moves freely through road network on its optimal path, respecting traffic rules.

The following agent classes were developed: *car*, *street*, *intersection*, *semaphore*.

The method Car::tick() determines the behavior of a car within simulation at each tick. It can be described with the following pseudocode:

```
attempt_changing_lanes;
if (at_intersection) decide_new_directon;
if (semaphore==red||semaphore==yellow) stop;
else if (distance_from_other_vehicle<ideal*K)
        speed -= k1;
else speed += k2;
do_move;
if (at_finish) self_destruct;
```

Function *attempt_changing_lanes* decides whether driver *wants* to change lanes, and whether he/she *can* do that (e.g. another car is in sight). *decide_new_direction* simply queries the shortest path determined at the time car was created – driver will remain oblivious to the state of traffic, as is usually the case (otherwise traffic jams wouldn't be such a big problem).

Note that coefficients K, k1 and k2 are not really constant (we just used a shorter notation), they are a function of current speed of *this* vehicle. Unlike other models, we assume that driver can't reliably determine other car speed, just the distance and own speed.

Figure 2 shows a screenshot of model execution.

An important issue that didn't receive the necessary treatment in literature is that of output metrics from simulation. In various what-if analyses there is a need for an output value helping us decide whether the change we made is improvement or not.

A number of papers mention *average trip time* as output metric. In our model the distances that vehicles pass vary with factor of 10 or more, therefore we believe that such metric isn't relevant. Thus we suggest *average speed* as an average of average speeds of all simulated vehicles, which in turn is the quotient of distance traveled and total time passed as the vehicle leaves simulation.

In case of very low congestion, a car not making any turns should reach close to its ideal speed. However, as cars turn and wait at semaphores, even very low congestion results in average speeds of at most 60-70% of ideal (see chapter on Measurements). As number of cars increases, this value further drops due to traffic jams.

It can be trivially shown that average speed relates to average throughput of network.

Another interesting value is *number of simulated vehicles* (not counting cars that have exited simulation). If this value grows constantly we can conclude that system isn't stable, that eventually there will be a total congestion.

Another metric to consider for future work is a value that would describe driver frustration due to driving slower then their desired speed. This can be expressed either as percent of time spent driving slower then ideal, or difference in travel time and expected time.

Further metric mentioned in literature are air pollution, throughput of given intersection or road, and likelihood of collision. Addition of such metrics to the model can also be considered

IV. Evaluation of Model and Example of Use

In the following text, we will demonstrate how described model can be used, in the example of comparing coordinated semaphore system to uncoordinated ones. We have attempted to determine whether "green wave" improves throughput of the

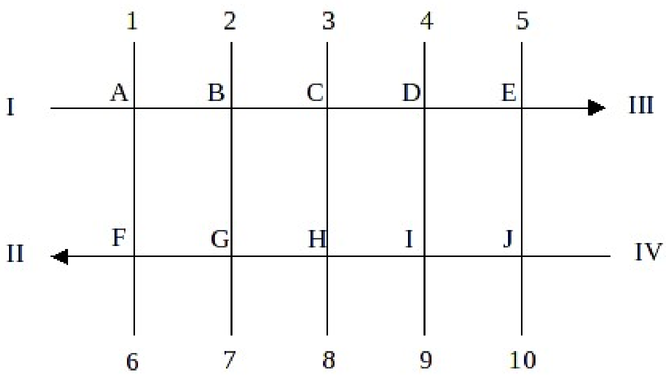

Figure 3. Scheme of model number 2

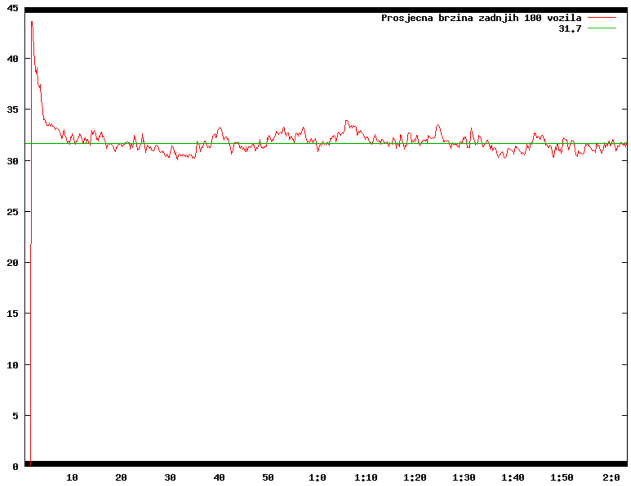

Figure 4. Average vehicle speed with model 3 and a coordinated semaphore system

traffic system by measuring average speed attained by vehicles that pass through the coordinated system. This result is compared to a system without coordination.

For the purpose of this simulation, we have created three models of roads in a hypothetical city:

1. First model consists of a one-way street 2km long, and five perpendicular street 1km each. These perpendicular streets cross the main one-way street at equal distances of 400 meters. Main street get most of the traffic: 10 cars per minute have entrance and exit on this street, while other possible relations get 0,5 cars/minute each. This results at the intersections being rather close to their projected saturation.
2. Second model, in addition to above, has another one-way street parallel but in opposite direction to the main street (see Fig. 3). Length of perpendicular is increased to 2km, so now the modeled area is 2x2 km. Distance between one-way streets is 660 meters. As both one-way direction now receive 10 cars/minute, the total number of cars in the simulation is somewhat larger.
3. Finally, in the third model, one-way streets are converted into two-way streets. For easier synchronization, these streets are prolonged to 4km, with intersections every 660 meters.

In each of these models, two different modes of semaphore timing were used:

1. All semaphores turn to green at the same time. (This in principle means that a car heading on main street stops at every, or at best every other semaphore.)
2. Coordination established using standard timing techniques. [16].

For each of these models a measurement of average vehicle speed is obtained, given "ideal speed" of 60 km/h which is also the speed for which semaphores are coordinated.

TABLE 1 MODEL TESTING RESULTS (AVERAGE SPEED)

Model	Average speed (km/h)		
	No coordination	Coordinated	Delta
One-way	30,3	33,0	2,7
Two one-ways	28,8	30,4	1,6
Two two-ways	30,6	31,7	1,1

To further analyze average speed fluctuations, we have added the ability to log current value of average speed during the simulation. To avoid the bias caused by older measurements, we have limited this value to last 100 vehicles to leave the simulation. Graph of these values can be seen on Fig. 4.

Since one of the issues of this paper is the performance of our model, we have measured the time required to simulate a given period (one hour).

The application developed allows user to specify execution speed in real-time factor. For example, speed of 10x means that the model should execute ten times faster then the real time, so simulating two hours of traffic should take 12 minutes. The emphasis here is on the word "should", as if the task is too computationally demanding, execution will be slower. We have observed that execution speed is determined primarily by number of simulated vehicles (agents).

We have tested this model on a laptop with Pentium M CPU running at 1.73 GHz with 2 MB L2 cache. Other features of execution machine such as graphics card or hard disk shouldn't affect the simulation. We have first executed the simulation at 10x speed and the timing revealed that there were no delays, e.g. one simulated hour took exactly 6 minutes. Therefore we increased speed to 20x which was too fast for the given configuration. The measured execution times for various models are given in the table below.

TABLE 1 MODEL TESTING RESULTS (EXECUTION TIME)

	Time to simulate one hour of traffic	Equivalent speed
One-way	5:05	11,8x
Two one-ways	03:38:00 AM	16,5x
Two two-ways	5:18	11,3x

There was no discernible difference in execution time between the uncoordinated and coordinated semaphore systems.

V. Conclusions and Future Work

In this paper we have presented state of the art in the area of agent-based traffic modeling. We have also presented a number of alternate solutions and papers in traffic simulation. Based on that knowledge, we have developed our own agent-based system. During development, we have made a number of assumptions with the primary goal being simplicity of development, but even with those assumptions the usefulness of such model has clearly been demonstrated.

Using this model we have confirmed a very specific prediction of microscopic level and detailed nature: namely, that coordinated semaphore systems result in an improved throughput of the road network (see Table 1). At the first look, delta values seem rather small. There are a number of factors to such result: stohastic nature of the simulation leads to periodic jams on certain intersections which reduce the green wave effect and rapidly decrease average speed (see Fig. 4). Also, vehicles moving outside the main route stop at the red light at least twice, and thus have disproportionally low average speed, which represents noise in the overall statistics. These factors could possibly be mitigated by a better choice of metrics, which remains to be researched.

Finally, we have proven that even on a low-performance laptop, the simulation executes at over 10x the real-time speed (see Table 2). Given that agent-based models are amenable to parallelization, the speed can further be increased if necessary. Note that the largest simulated area was 2x2 km, and the density of streets in that specific model was rather low (although those streets were fairly congested). As new features and abilities are added to the model, some decrease in execution speed is to be expected.

Future work should first and foremost be directed towards improving the model presented in this paper. One possible road to be taken is attaining detailed traffic measurements performed in the city of Sarajevo, and comparing them to outputs from a model representing part of the actual Sarajevo road network. Also, the model should be improved with support for variable vehicle length (e.g. trucks and buses), internal traffic (parking lots), and complex road topologies (roundabouts etc.)

Pending the above, there are a number of future research directions. One promising area which is poorly researched in current literature is the issue of output metrics from traffic simulations, as discussed here in chapter III. A number of test could be conducted, evaluating various metrics and their correspondence to desired values of a real-life system.

Also, this model could be used for testing feasibility of "intelligent semaphores": given the cheap and ubiquitous computing, in addition to communication between semaphores, future semaphores could use complex decision logic to determine the exact timings.

References

[1] Eric Bonebau, Agent-based modeling: Methods and techniques for simulating human systems, *Proceedings of the National Academy of Sciences of United States*, 2002. Retrieved from: http://www.pnas.org/content/99/suppl.3/7280.full (11.Jan 2009)

[2] Charles M. Macal, Michael J. North, *Tutorial On Agent-Based Modeling and Simulation Part 2: How To Model With Agents*, Winter Simulation Conference '06. Retrieved from: http://www.informs-sim.org/wsc06papers/008.pdf (11.Jan 2009)

[3] Burmeister B., Haddadi A., Matylis G., "Application of multi-agent systems in traffic and transportation", IEE Proc.-Softw. Eng., Vol 144, No. 1, February 1997.

[4] Wardrop, J. G., 1952. Some theoretical aspects of road traffic research, *Proceedings, Institute of Civil Engineers*, PART II, Vol.1, pp. 325-378.

[5] Rongui Liu, Dirck Van Vilet, David P. Watling, "DRACULA: Dynamic Route Assignment Combining User Learning and Microsimulation", Paper Presented at PRTC, 1995, Vol. E, pp. 143-152. Retrieved from: http://www.its.leeds.ac.uk/software/dracula/downloads/Paper17-Paper%20presented%20at%20PTRC.pdf (4.Jan 2009)

[6] Martin Fellendorf, Peter Vortisch, *Validation of the Microscopic Traffic Flow Model VISSIM in Different Real-World Situations*, Karlsruhe, Germany, 2001. Retrieved from: http://cgi.ptv.de/download/traffic/library/2001%20TRB%20VISSIM%20Validation.pdf (4. Jan 2009.)

[7] Kutluhan Erol, Renato Levy, James Wentworth, *Application of Agent Technology to Traffic Simulation*, Federal Highway Administration (FHWA). Retrieved from: http://www.tfhrc.gov/advanc/agent.htm (3. Jan 2009)

[8] Roland Chrobok, Sigurður F. Hafstein, Andreas Pottmeier, "OLSIM: A New Generation of Traffic Information Systems", *Forschung und wissenschaftliches Rechnen*, University Duisburg-Essen, 2004. Retrieved from: http://www.billingpreis.mpg.de/hbp03/OLSIM.pdf (10. Jan 2009).

[9] K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic, *J. Phys. I France* 2 2221-2229 (1992).

[10] Esser J., Schreckenberg M., "Microscopic Simulation of Urban Traffic Based on Cellular Automata", *Int. J. Mod. Phys* C8(5), 1997.

[11] Kai Nagel, Marcus Rickert, "Parallel implementation of the TRANSIMS micro-simulation", *Parallel Computing,* vol. 27, issue 12, November 2001, pp. 1611-1639. Retrieved from: http://arxiv.org/pdf/cs.CE/0105004 (11.Jan 2009)

[12] Burmeister B., Haddadi A., Matylis G., "Application of multi-agent systems in traffic and transportation", *IEEE Proc.-Softw. Eng.*, Vol 144, No. 1, February 1997.

[13] P.A.M. Ehlert, L.J.M. Rothkrantz, "Microscopic traffic simulation with reactive driving agents", *2001 IEEE Intelligent Transportation Systems Conference Proceedings.*

[14] S. Panwai, H. Dia, "A Reactive Agent-Based Neural Network Car Following Model", *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, Vienna, Austria, September 13-16, 2005.

[15] Kai Nagel, Paula Stretz, Martin Pieck, Shannon Leckey, Rick Donnelly, Christopher L. Barret, *TRANSIMS Traffic flow characteristics*, 21 Oct 1997. Retrieved from: http://arxiv.org/pdf/adap-org/9710003v1 (11. Jan 2009)

[16] *NIATT Signal Timing Tutorial*, National Institute for Advanced Transportation Technology, University of Idaho. Retrieved from: http://www.webs1.uidaho.edu/niattproject/ (4. Jan 2009)

[17] Julian Laufer, "Freeway capacity, saturation flow and the car following behavioural algorithm of the VISSIM microsimulation software", 30th Australasian Transport Research Forum, 2007. Retrieved from: http://www.patrec.org/atrf/papers/2007/1627_Laufer_Freeway%20capacity,%20saturation%20flow%20and%20car%20following%20behavioural%20algorithm%20of%20VISSIM%20microsimulation%20software.pdf (4. Jan 2009)