HEAD LEADING ALGORITHM FOR URBAN TRAFFIC MODELING

David Hartman
Department of Computer Science and Engineering
University of West Bohemia
Univerzitní 22, Plzeň, 360 14, Czech Republic
Email: tazman@kiv.zcu.cz

KEYWORDS

Simulation, Urban Traffic, Cellular Automata Model, Nagel-Schreckenberg model, Head Leading Algoritm, JUTS project.

ABSTRACT

This paper describes construction of discrete urban traffic simulation model based on Nagel-Schreckenberg's cellular automata model (NaSch model) and its modification. These NaSch models have been constructed for simulation of traffic on highways and freeways. We try to modify this aproach to obtain a model of urban traffic. The development is based on an object oriented approach.

GOALS OF THE MODEL

Our objective was construction of a model that could be used for analysis of traffic situation in urban areas, i.e. jams, queues of vehicles, public transport programmes, junction lights switching algoritms, etc. Another very important requirement was object oriented character of the designed model, because we wanted to have the possibility of easy or half-automatic creation of simulation map. It also means that simulation map should be divided into parts (representing urban traffic entities), that are connected together and the whole simulation is performed by communication among them.

CHOOSED MODEL TYPE

According to the great study of traffic flow modeling in TBR report (Gartner et al. 1997) and the great study of micro-simulation models in European traffic simulation project SMARTEST (Alergs et al. 1998), we decided to use a micro-simulation model as our basic model (instead of macroscopic or mesoscopic types). We choosed the Nagel-Schreckenberg cellular automata model, because of its detailed simulation and also easy implementation and natural character of vehicle move modelling. But there was one disadvantage. This model was originally designed for highway or freeway traffic (Nagel and Schreckenberg 1992). So it was important to make some adjustment to ensure that the simulation of urban traffic will be also possible. There were other approaches to do that. An example can be found in the Simon and Nagel simplified model (Nagel and Simon 1998). They used a simplified version of street network and just single-lane simulation. A More detailed model is constructed by Esser and Schreckenberg (Esser and Schreckenberg 1997). Other approaches can be found in other papers like (Chrobok et al. 2001), (Schreckenberg et al. 2001) and (Chopard et al. 1997). In our case a strong combination with object oriented approach is used. We modified the basic NaSch traffic model and constructed the object oriented model of traffic network that can use a modified cellular automata model as its basis. And finally we added the leading head algoritm into the global model to make the modification in NaSch and communication in object model possible. So the leading head algorithm is a micro-simulation algorithm and also the algorithm for object oriented model control.

BASIC NAGEL-SCHRECKENBERG MODEL

In this section we describe the basis of the basic NaSch cellular automata model. More detailed description can be found in other papers (Knospe et al. 2001) or original model definition in paper (Nagel and Schreckenberg 1992).

To ensure the completeness of our paper we explain the basis of this model. Let's have a look on the basic NaSch model concerning a single lane situation only. In this model the road is divided into cells. Each cell has the same length of 7.5m. Each cell can be occupied by a single vehicle. The vehicles are therefore just like items of an array. Each vehicle has a discrete speed that represents the number of cells that the vehicle jumps over during its movement (see Figure 1).

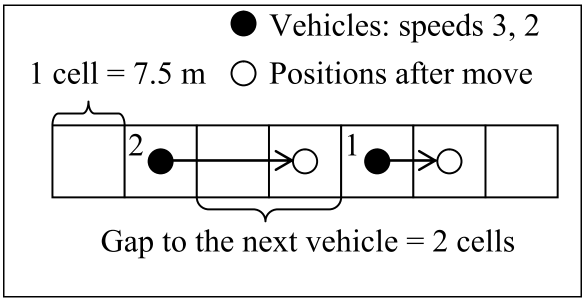

Figure 1: Road Division in NaSch Model

The movement is performed by steps of simulation. During each step the vehicles are shifted about the number of cells that equals their speeds. The positions of the vehicles together with vehicles states constitute the global state of the model at a specific discrete time *t*. Let *i* be an index of the state (i.e. the value of discrete

Proceedings 16th European Simulation Symposium
György Lipovszki, István Molnár © SCS Press, 2004
ISBN 1-56555-286-5(book) / ISBN 1-84233-106-x(CD)

time t). If we denote the speed of the vehicle as v_i and gap to the next vehicle as g_i we can obtain next state s_{i+1} from the actual s_i by applying the following basic rules of NaSch model:

1. Acceleration: $v_i \rightarrow \min(v_i + 1, v_{\max})$
2. Breaking: $v_i \rightarrow \min(v_i, g_i - 1)$
3. Randomization: $v_i \rightarrow \max(v_i - 1, 0)$ with p
4. Driving: $x_i \rightarrow x_i + v_i$

Here we can see the roots of the model. The drivers of the vehicles want to accelerate as much as possible (step n. 1), but they have to slow down to avoid crashes (forbidden in the model). So the speed is restricted by gap to the next vehicle (step n. 2). Then the vehicle can move (step n. 4), but for realistic behaviour of the model the 3rd step is added. The adjustment of speed at step n. 3 is performed only with probability p. This stochastic step (others are deterministic) takes into account the natural velocity fluctuation (human factor or road conditions).

Now we can look at the multi-lane case. We simply take single lanes and place each one alongside the other and add the lane changing rules (see Figure 2) to the dynamic mechanism.

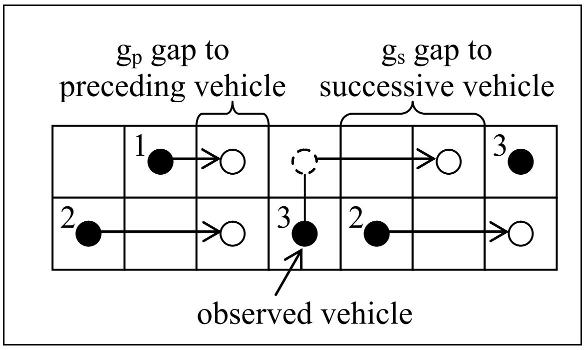

Figure 2: Lane Changing Situation

The lane changing rules differ in used models but their principles are often based on checking the values of g_s and g_p. A systematic approach for two-lane rules can be found in (Nagel et al. 1998) and also good information about this problems can be found in (Knospe et al. 1998), but mainly for highways.

When we use these rules, the simulation step is divided into two sub-steps. The first one perform the lane change manoeuvres and the second one perform the NaSch rules. As we see (Figure 2), the vehicle after applying lane change rules is in a position that is the same as in a normal state, so you can easily perform the standard NaSch rules without any problem. There could be some problems when you use the vehicles with length more than one. We discuss it later when we describe our model that uses this approach. For another solution of this problem, see paper (Esser and Schreckenberg 1997). Lane change manoeuvres are also used for modeling overtaking. There could be also problems with deadlocks during lane changing. Deadlocks occur when the vehicle that wants to change the lane waits until the target lane is free and another vehicle waits for it and blocks it together.

INCLUDING SLOW-TO-START BEHAVIOUR

A very important part of the model is the stochastic step n.3. The basic NaSch rules use a constant probability for this step, but it seems to be insufficient for modeling metastable states with a very high flow (Barlovic et al. 1998). These metastable states relate with restart behaviour of stopped vehicles. To include this behaviour to simulation in more realistic fashion new models with slow-to-start rules are introduced, e.g. T^2, BJH and VDR model (Barlovic et al. 1998).

The Velocity-Dependent Randomization (VDR) model is based on the idea of dependent probability. It's simple. The probability in step n.3 from NaSch rules is a function of the vehicle's speed, i.e.

$$p = p(v(t))$$

The parameter p should be determined before the original 1st step from the NaSch rules. Sometimes, often for explanation, is used a very simple case of this model with probability function with this definition:

$$p(v) = \begin{cases} p_0 & for \quad v = 0 \\ p & for \quad v > 0 \end{cases}$$

Using one of the mentioned models introduces the restart behaviour into the model and ensures that traffic flow characteristics will better fit realistic values.

ANTICIPATION IN CELLULAR AUTOMATA

Last modification of NaSch cellular automata model is based on introducing anticipation of the following vehicle's movement. It means that an anticipation rule is added to the original set of rules. It is also useful to add this rule to the VDR model instead of the original NaSch model, because the slow-to-start rule of the VDR model still helps the simulation to be more realistic in modeling restart behaviour and therefore metastable states.

The anticipation rule is a modification of the original adjustment of speed. This new adjustment takes into account the expected behaviour of the leading vehicle (following vehicle in the direction of move). This rule helps the model to simulate different vehicle characteristics like acceleration more realistic. It is also introduced in cases of different vehicle length. In these cases the length that represents one cell is usually set to a smaller value, e.g. 1.5 m (Knospe et al. 2001). The next advantage of anticipation is speed-up effect for lane changing modeling caused by reduction of changing wait times. For systematic approach and more detailed description of this problem including the lane change problem, see (Knospe et al. 1998).

BASIS OF MODEL AND CELL LENGTH

The model that we constructed is based on NaSch cellular automata as we mention above. We also used a VDR and anticipation modification of the basic algorithm. The original NaSch model is designed for highways and freeways and if we use 1 second as simulation step (as usual) the speeds of the vehicles follow the Table 1.

Table 1: Original NaSch Speeds

Discrete speed [cell/s]	Real speed [km/h]
0	0
1	27
2	54
3	81
4	108
5	135

Our model is intended to be used for urban traffic and therefore these speeds are unusable (few values with respect to permitted speed in cities) and also cell length is very long for city and we want to distinguish individual types of vehicles (mainly vehicle length). These requests result in cell length of 2.5 m. For the permitted speed of 50 km/h used in the Czech Republic, the speed will be as shown in Table 2.

Table 2: Modified Speed Discretization

Discrete speed [cell/s]	Real speed [km/h]
0	0
1	9
2	18
3	27
4	36
5	45
6	54

This length ensures a sufficient number of needed speed values (Table 2) and also represents approximately all known vehicles as shows the Table 3.

Table 3: Representation of Types of Vehicles

Vehicle length [cell]	Real length [m]	Type of Vehicle
1	2.5	motorcycle
2	5	passenger vehicles
3	7.5	van
4	10	minibus
5	12.5	bus and truck
6	15	lorry

These types of vehicle will be used in our discrete model.

TRAFFIC NETWORK STRUCTURE

We know the type and modifications of the NaSch model and now we can deal with the traffic network structure, because it will be very important for leading head algorithm construction. The whole map definition is strongly object oriented. Most of the traffic network elements are objects.

Road Segment

We start from the basic element of network. The modified NaSch algorithm is defined for set of traffic lanes in one direction and therefore the road segment is defined as a set of these lanes (Figure 3).

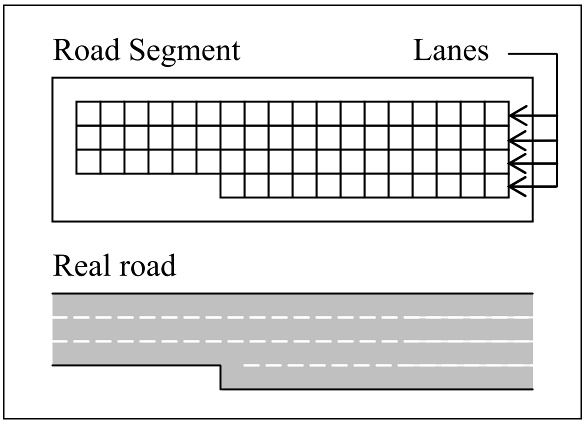

Figure 3: Road Segment

This segment is the basic one, because the main part of simulation is performed here.

Crossroad Segment

Another important segment is the cross-road. Because the dynamics at these segments are very complicated, there must be an other mechanism than NaSch model to perform vehicle movement. But we also wanted to build up this mechanism on very similar principles. Finally we meet our requirements by introducing special single places that communicate with each other. Through these places the vehicles pass through the crossroads. Figure 4 shows a simple crossroad's structure.

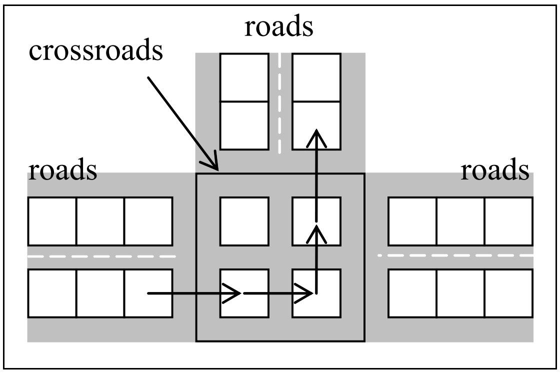

Figure 4: Simple Crossroad Structure

The crossroad solutions are very different in the field of cellular automata models. Sometime just a switching matrix with probabilities is used (Esser and Schreckenberg 1997), or a representation by a rotary's dynamic (Chopard et al. 1997) or by structure from ramps and transfer links (Chrobok et al. 2001).

Roudabout Segment

Now we have roads and crossroads, so we can make a traffic network as an oriented graph from roads as edges and crossroads as nodes. But there exists another important segment except these two. It is the roundabout and it is also a node in our oriented graph. We have two possibilities how to define this object. The first is to define it same way as a crossroad, but if we construct the model of a large roundabout, the number of places will exceed a reasonable value. Therefore the second possibility will be better and that is to construct the roundabout as a road segment that ends at its beginning. We can do that because we can join the onramps and offramps to a road, so we can also join incoming and outgoing roads to the roundabout. The structure is shown in Figure 5.

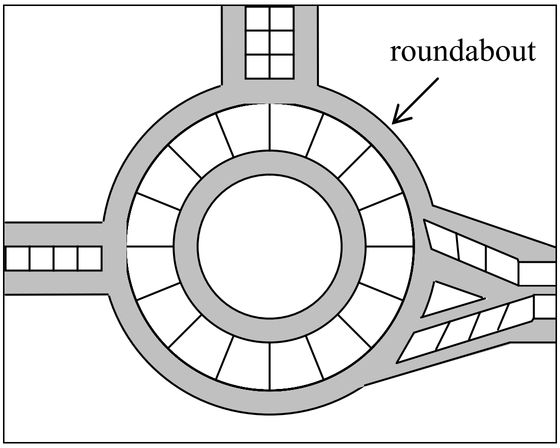

Figure 5: Crossroad structure

Now it seems that we have all segments we need to build up the oriented graph that represents the structure of the traffic network. But there are still one or two segments more to make the network complete.

Generator and Terminator Segments

To include the environment of the simulated system we need to create segments that will generate vehicles following some stochastic distribution and send those to the network. This distribution differs from place to place (it means from road to road), so there should be more generator segments that should be connected at a specific place to the network.
Except generation of vehicles we probably need a segment to terminate the life of a vehicle. This segment can collect some statistics and check for problems.

Connecting Segments by Accessplace

We have now defined several segments of the traffic network. We know that these segments should be connected together. A connection can be between a road and a crossroad, a road and a generator, a road and a road, etc. So many different segments can be connected together in a pair. It will be usefull to find the way how to connect these segments by general mechanism. We establish this mechanism as a connection by an Accessplace. An Accessplace is a special object that knows inner structure of the participating segments. With this knowledge it can send vehicles from any segment to any other without condition that these segments must know their neighbours.
The final structure of the traffic network implemented in our model for the case of crossroad shown in Figure 4 is shown in Figure 6.

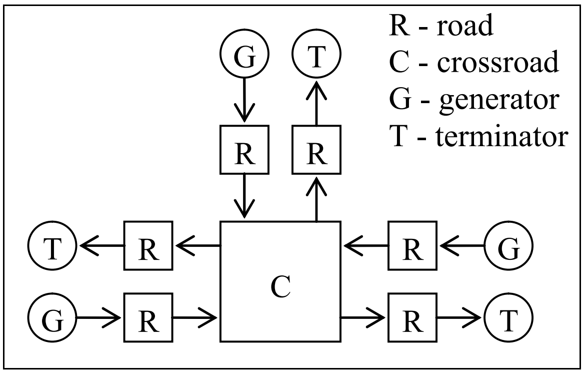

Figure 6: Traffic Network Structure

The structure of the simulation map in Figure 6 is shown without Accessplaces that will be between each pair of connected segments.

LEADING HEAD ALGORITM

We have defined the structure of the traffic network and we have defined each segment of this structure. Now it's time to describe our leading head algorithm (LHA) for which this structure was built. The basic principle of leading head algorithm is simple. We assume that the vehicles should be with length more than 1. Without this assumption the LHA has no meaning. The movement of the vehicle follows these steps (when the LHA is used):

1. The head of the vehicle moves as a standard vehicle in the NaSch model.
2. Other pieces of the vehicle are shifted over the same way that the head was shifted before.

The principle of the algorithm is shown in Figure 7. The important thing to notice is that testing possible collisions is needed only for the head and the following pieces move automatically without testing. These pieces only occupy the cells on the road. The LHA is also a good mechanism for sending vehicles from one segment to another, because the algorithm only concerns the head and the other parts of vehicle that move over the signed path.

So they need not to pass through the communication object in such a complicated way that the head passes. One can say that the LHA is an algorithm for hiding the length of the vehicle.

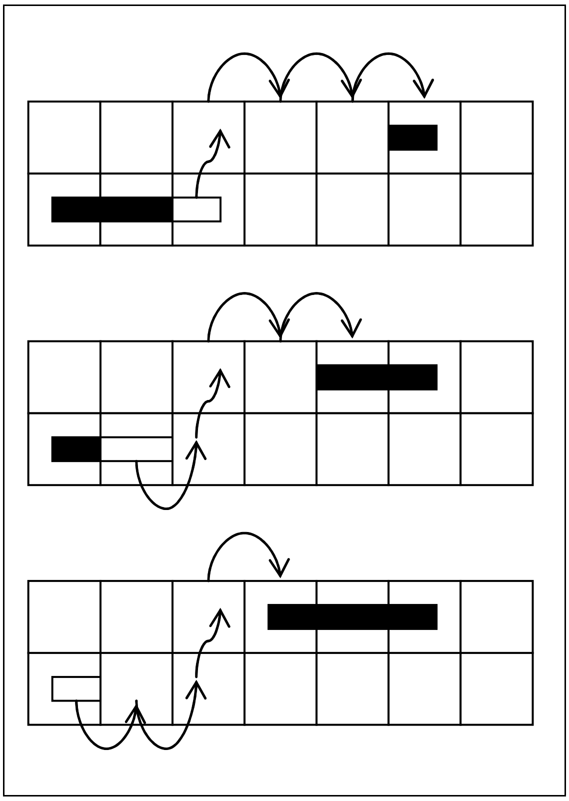

Figure 7: LHA Lane Change

The last advantage of the LHA algorithm together with the designed traffic network is an easy way of a vehicle passage through a crossroad. The pieces of the crossroad are also possible positions of the vehicle, so if the head is going through the path at the crossroad, the other pieces of the vehicle can follow it without problem. This situation with just the head move (the other pieces' moves could be easily added) is shown in Figure 8.

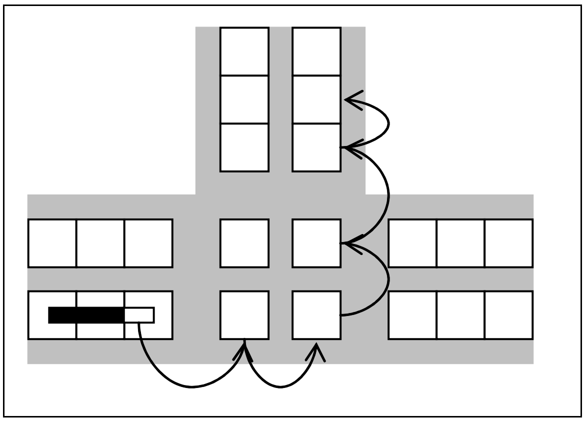

Figure 8: LHA at Crossroad

We have seen the LHA in action during lane change process, we have talked about communication between segments with a help of LHA and we have seen the LHA solution of crossroad passage. Finally we have to say that this algorithm probably doesn't bring any acceleration of simulation run, but it uses a general method for vehicle moving that could help us to extend possible developed models.

CASE STUDY

This type of NaSch cellular automata built-in traffic network model, together with lane head algorithm is being developed in the JUTS project (J-Sim Urban Traffic Simulator). This project is written in Java and uses all these principles inside its simulation core. We are trying to create a complex simulation tool with easy methods for editing and generating the simulation map. We are using XML format to store and work with the simulation map and other data.

But until these days the core has not been validated yet, because it is still under development. For more information about this project see (Hartman and Kačer 2003).

We cooperate with Public Transport Department of Pilsen (a Czech city) from which we obtain data from several observers that we want to use for half-automatic simulation map generation. For that purpose the designed model is well suited.

CONCLUSION

The designed model is based on the Nagel-Schreckenberg cellular automata model with VDR and anticipation that have been tested several times in other projects or smaller research works. So there is quite great guarantee that the simulation based on this model will approach reality.

The solution of crossroad dynamics is one of the most detailed that is used in cellular automata models and at the same time it allows a very simply way to pass the vehicles through the crossroad. It involves a possibility of changing the inner dynamics by changing the permitted directions on the crossroad or you can simply add any new testing of other interesting features during passage through the places at the crossroad.

The leading head algorithm is an easy and flexible way how to move vehicles through a segment and also between segments. The crossroad and road are mostly the same for this algorithm and also the roundabout which is designed as a road in cycle is the same.

Finally we must say that the main goal of the designed model is to make an easily extendable model for big applications that will be able to configure the simulation automatically from the urban traffic data that is obtained from city measurements.

ACKNOWLEDGEMENT

The research was supported by a grant of the Grant Agency of the Czech Republic - Research of methods and tools for verification of embedded computer systems, no. 102/03/0672.

USEFULL LINKS

http://www.its.leeds.ac.uk/projects/smartest/
- SMARTEST project (Simulation Modelling Applied to Road Transport European Scheme Test) review micro-simulations.

http://www.trb.org/
- TBR home page (Transportation Research Board).

http//www.traffic.uni-duisburg.de/
- Group “Physics of Transport and Traffic” of prof. M. Schreckenberg.

http://www.juts.zcu.cz/
- JUTS project (J-Sim Urban traffic Simulator) home page.

REFERENCES

Algers S.; E. Bernauer; M. Boero; L. Breheret; C. D. Taranto; M. Dougherty; and K. F. and J.-F. Gabard. 1998 “A Review of Micro-Simulation Models.” SMARTEST Project report, Institute of Transport Studies Univerzity of Leeds, UK.

Barlovic R.; L. Santen; A. Schadschnaider; and M. Schreckenberg. 1998 “Metastable states in cellular automata for traffic flow.” *European Phys. Journal*, B 5, 793-801.

Chopard B.; A. Dupuis; and P. Luthi. 1997 “A cellular automata model for urban traffic and its application to the city of Geneva.” *Proceedings of traffic and granular flow ‘97*, Duisbourg, 154-168.

Chrobok R.; J. Wahle; and M. Schreckenberg. 2001 “Traffic forecast using simulations of large scale networks.” *4th International IEEE Conference on Intelligent Transportation Systems*, Oakland, pp. 434-439.

Esser J. and M. Schreckenberg. 1997 “Microscopic simulation of urban traffic based on cellular automata.” *International journal of modern physics*, Vol. 8, 1025-1037.

Garthner N.; J.C. Messer; and K.A. Rathi. 1997 “Traffic Flow Theory.” Update and expansion TRB Special Report 165, Federal Highway Administration, U.S.Department of Transportation.

Hartman D. and J. Kačer. 2003 “JUTS – J-Sim Urban Traffic Simulator.” *Proceeding of the 2nd international Conference on the Principles and Practice of Programming in Java*, Kilkenny, Ireland, 113-116.

Knospe W.; L. Santen; A. Schadschnaider; and M. Schreckenberg. 1998 “Disorder effects in cellular automata for two-lane traffic.” Psysica Journal A 265, 614-634.

Knospe W.; M. Schreckenberg; R. Barlovic; and H. Klüpfel. 2001 “Statistical physics of cellular automata models for traffic flow.” *Computational Statistical Physics*, Springer, 113-126.

Nagel K. and M. Schreckenberg. 1992 “A cellular automaton model for freeway traffic.” *J. Phys I France 2*, 2221.

Nagel K.; D.E. Wolf; P. Wagner; and P. Simon. 1998 “Two-lane traffic rules for cellular automata: A systematic approach.” *Physical Review E*, American Physical Society.

Schreckenberg M.; L. Neubert; and J. Wahle. 2001 “Simulation of traffic in large road networks.” *Future generation computers 17*, 649-657.

Simon, P.M. and K. Nagel. 1998 “Simplified cellular automaton model for city traffic.” *Physical Review E*, The American Physical Society.

AUTHOR BIOGRAPHIES

DAVID HARTMAN was born in Karlovy Vary Czech Republic and went to the University of West Bohemia, where he studied software engineering and obtained his degree in 2003. Than he started PhD. studies at the Departement of Computer Science at this university and currently works on the simulation problems. He is a member of the research group DSS (Distributed Systems, Software engineering and Simulations), creator and one of the leading members of the traffic simulation project JUTS. His e-mail address is: `tazman@kiv.zcu.cz` and Web-page is: `http://www.kiv.zcu.cz/~tazman/`.