MAS Model of the Level Crossing

Ales Janota
Department of Control and Information Systems, Faculty of Electrical Engineering, University of Zilina
(Univerzitna 8215/1, SK-010 26 Zilina, Slovak Republic, Tel.: +421-41-5133356, E-mail: ales.janota@fel.uniza.sk)

The paper introduces a new unpublished version of the multi-agent model of the level crossing system and summarizes research work that has been done since release of the previous concept model in late 2005. The multi-agent approach allows effective modelling of the complex control process of traffic flows from two different kinds of transport. The model respects basic function safety requirements and its graphic design truly corresponds to the Slovak/Czech-like road-rail interface. Code of the model was designed using the hybrid compiler/interpreter NetLogo 3.1.4. Through different initial settings traffic flows for railway and/or road traffic may be defined and different parameters calculated and observed based on simulation. Partial attention is paid to brief survey of newly implemented functions. Within conclusions the author also presents some further improvements and intentions of the future research.

Keywords: *Multi-Agent System, Level Crossing, Traffic Flow, Function Safety Requirements, NetLogo*

1. Introduction

The multi-agent system (MAS) is generally composed of several autonomous, interacting entities – agents [1]. Individual agents can be more or less intelligent, however collectively they are capable of reaching goals that are difficult for one agent and so manifest self-organization and complex behaviour. MASs are particularly well suited for modelling of complex systems developing over time. Traffic can be viewed as a complex system [2]. Possibility of giving instructions to hundreds or thousands of independent "agents" all operating concurrently makes it possible to explore the connection between the micro-level behaviour of individuals (e.g. road vehicles, rail vehicles, components of a level crossing system) and the macro-level patterns (objective function of the level crossing system) that emerge from the interaction of individuals. Developing macro models is one of the primary approaches to modelling complex systems. Macro models focus on the observable behaviour of a system and define it in terms of aggregate, abstract parameters. In the case of traffic, they are usually derived from fluid dynamics. As a sort of successful applications in road transport domain many examples could be mentioned, e.g. a hierarchical MAS for solving very complex and highly interactive problem of urban traffic optimization [3], MASs for traffic light control and regulation [4][5][6] and others. Positive examples could also be found in the railway transport domain [7].

The first idea of how the MAS approach could be utilised in modelling of level crossing systems was brought into life in late 2005 when the initial concept model was published [8] and the code was made available within NetLogo community models [9]. That work represented the first study that implied a lot of simplifications and became an introductory concept designed to test applicability of the MAS approach to the given problem. In recent time more effort has been made to obtain an essentially improved model, offering more realistic view based on advanced graphic representation and behaviour simulating most of really implemented functions. The paper is intended to sum up the results of this research work.

2. NetLogo model of the level crossing

The model discussed in the paper was developed and tested in the hybrid compiler/interpreter NetLogo 3.1.4 which is a programmable modelling environment primarily designed for simulating natural and social phenomena [10]. The interface window of the model is shown in Figure 1. One of the long-time objectives of performed work is design of a set of EU national road-rail interfaces implemented and selectable as an optional parameter of the model. This approach would help to better understand level crossing types and technologies operated by different national road/railway operators and identify equalities and differences as a necessary condition of the future harmonization (this process has already been initiated within the European SELCAT project 2006-2008, [11]). Visual appearance of the model presented in the paper is based on real level crossing systems operated at Slovak and Czech railways (ZSR, CD). For the sake of simplification the railway traffic is considered as single line traffic (with trains moving only in the West-East direction, i.e. from the left to the right of the simulation window).

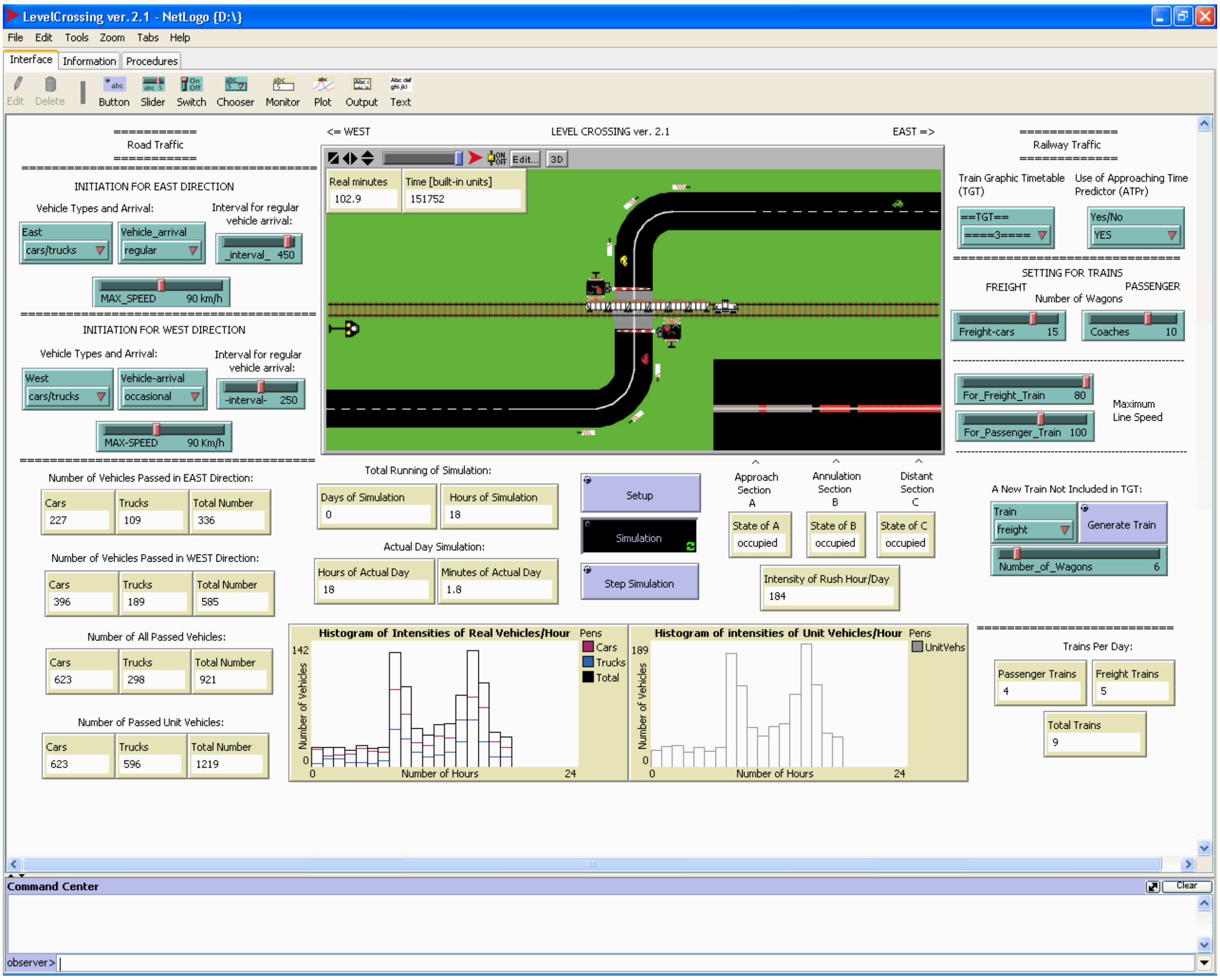

Figure 1. Interface window of the model (Example of Slovak/Czech road-rail interface)

2.1. Implemented functions

From functional point of view each level crossing system may be divided into the road and railway part. These parts are functionally separated but in safety protection they effect in coherence with each other. The structure of such a functional platform is indicated in Figure 2.

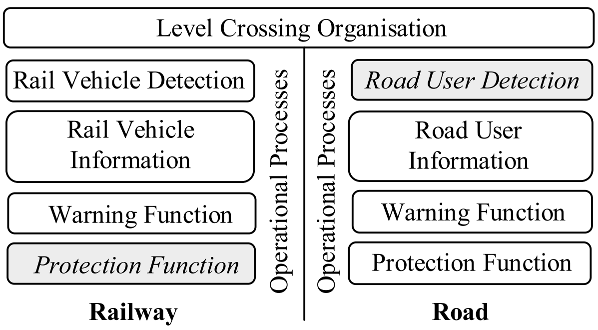

Figure 2. General structure of the functional platform

This structure is to be used for modelling of all identified functions, potentially realizable by different technologies. Since the real Slovak/Czech level crossing systems do not make use of a complete set of indicated functions yet, italics inside grey boxes accentuate two functions intentionally omitted in the model. At the road side the model implements the functions:

a) Informing a road user on the actual status of the level crossing: national speciality of the Slovak and Czech level crossings is optional existence of the so called "active signalling", represented by white flashing light saying that no rail vehicle is approaching or retreating the crossing. This kind of signal (if installed and activated) allows road users to cross the railway line at the higher speed 50 $km \cdot h^{-1}$; otherwise the lower speed limit 30 $km \cdot h^{-1}$ is appointed; in addition passive information signs are used to inform road users about existence of the level crossing ahead (situated in distances 80 m, 160 m and 240 m from the level crossing border). Each level crossing must be equipped with St. Andrew's crosses;

b) Road user warning is realized through two red alternatively flashing lights accompanied by acoustic signal (normally generated by the horn or bell);

c) Road user protection is realised by full- or half-barriers (no barriers means no protection).

The general rule resulting from national legislation says that the time between start of the warning period and the moment of train coming to the level crossing must be so long that the longest (22 m) and slowest (5 km.h^{-1}) road vehicle, situated inside a dangerous zone at the moment of warning state activation, is able to leave safely. At the railway side analogical functions can be seen:

d) Rail vehicle detection is in practice based on the use of track circuits or axle counters monitoring conditions of particular sections (approaching, annulation and distant sections). Activation of the warning state is derived from detecting a train in the approach section. Since speed of rail vehicles may vary for different kinds of trains (for example fast and slow passenger trains and/or freight trains), generally this fact has a negative effect on duration of time the road traffic participants must wait for train coming. Therefore, the additional function imitating existence of the *Approaching Time Predictor* (*ATPr*) is implemented in the model as an optional choice. If used, activation of the warning state depends on velocity of the approaching rail vehicle (generally, the slower train, the later activation).

e) Informing and warning a crew of the rail vehicle is given through the rail signal. In operation of Slovak and/or Czech railways this signal (if installed) consists of two yellow permanent lights arousing train crew‘s notice (sometimes substituted with yellow reflectors) and one white light being on or off, representing system being in operation or failed.

2.2. Graphical user interface

The graphic window makes the 2D "world" of the model visible. It is divided up into a grid of *patches* having coordinates `pxcor` and `pycor` (Figure 3). The patch coordinates `(0,0)` are always situated in the centre of the world - this default setting cannot be changed by the user, unlike the size or dimensionality (2D, 3D) of the simulation world [10]. Here the total number of patches (81×37=2297) is determined by the settings `screen-edge-x` and `screen-edge-y`. Agents representing both stationary and dynamic objects (e.g. cars, trains, traffic signs, barriers etc.) that "live" their lives in this world are represented by so called *turtles*, having coordinates `(xcor, ycor)`. Shapes of individual turtles-agents have been selected from a default library, imported from another models, or newly created and customized exploiting a built-in shape editor.

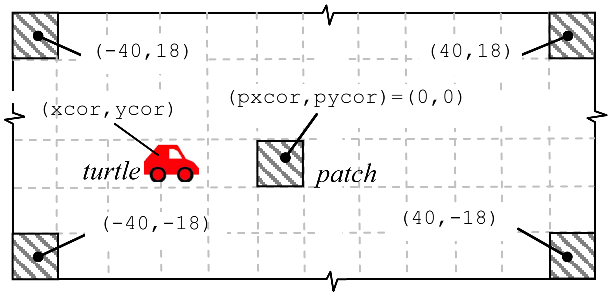

Figure 3. Two-dimensional NetLogo world

Majority of traffic-oriented NetLogo models that have been published so far depict road networks as a simplified network of rectangular lines (more detailed survey of traffic-oriented NetLogo models was presented in [8]). A lot of work has been done to make a graphical user interface more realistic in comparison with the previous concept model [9]. The essential part of the code responsible for drawing a part of the road curve is shown below. The meaning of particular procedures appearing in this sample code can be found in documentation available at [10].

```
to curve
   ask turtles [set heading 90 part1]
   …
end
to part1
   setxy(-1*max-pxcor+34)(-1*max-pycor+5)
   set pen-size 39
   set color black
   pd repeat 7 [fd 1.16 lt 12]
   pu
end
```

However, more realistic scene on one side brings a need to control turtles-agents movement in a more advanced way to ensure road trajectory following (e.g. pseudo-fluent rotation of road vehicle shapes according to their actual position on the road).

2.3. Optional settings

There are a number of new optional settings designed to make initialization of the model possible and to parameterize simulation running. To set road traffic properties, a user may use the control elements (buttons, sliders, switches, choosers) located in the left third of the interface window (Figure 1). The group of control elements situated in the right third of the interface window is dedicated for initialization of railway traffic.

All settings become functional after pushing the button *Setup* situated below the world. Simulation may be performed as a continuous or step-by-step simulation. Since the simulation program shows how traffic situation develops in time, the code contains definition of simulation time-units. Its actual value is displayed in the left-hand top corner of the world together with the value of real time derived from the internal system clock.

By customization of time-units (through modification of the code) it is possible to define how long one "simulation day" should really last. In addition to visualization effects, the model may also work with sound effects imitating audio warning.

2.3.1. Road traffic settings. Road vehicle flows may be defined independently for *West* and *East* traffic directions. The user may define a type of generated vehicles (cars, trucks, or random mixture of both cars and trucks). Another chooser element makes possible to choose the way of how these vehicles appear in the model – either based on regular arrival (once per defined time interval) or based on occasional arrival utilizing the quasi-Poisson distribution ensured by the built-in random generator (realized through the *random-exponential* procedure). Turtle-agents acting as road vehicles ("road vehicle agents") are prevented from occupying the same position, i.e. one turtle overlaying another one. To make simulation more realistic, road vehicles get colours randomly, too. Maximum speed limits are adjustable separately for both traffic directions. When reaching the edge of the world, mobile turtle-agents are being killed (deleted) to avoid their automatic appearance on the opposite side of the world (default property of the NetLogo environment).

2.3.2. Railway traffic settings. There are two kinds of trains to be generated – freight trains and passenger trains (slow trains and fast trains). Their length may be set through definition of a number of wagons (freight-cars or coaches). To simulate different speeds, it is possible to choose different speed limits for all kinds of trains. In a standard way trains are generated based on data included in the source code of the model, in the part called *Train Graphic Timetable* (TGT). The model actually contains three different TGTs. Figure 4 shows example of how TGT No. 1 has been defined.

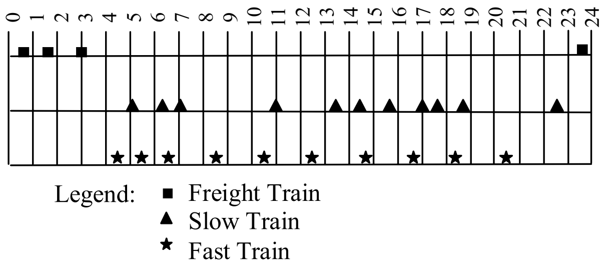

Figure 4. Example of Train Graphic Timetable (No. 1)

What's more, at any time a user has a chance to generate an extra train of the chosen type and length that is not included in the TGT. The only restriction is applied – at the moment maximally one train may be found in the simulation world.

2.3.3. Level Crossing system settings. From the viewpoint of level crossing operation there is possibility to enable function of the *Approaching Time Predictor* (*ATPr*). Its control element (chooser) is situated in the right-hand top corner of the interface window. According to [12], the basic task of the *ATPr* is to measure actual velocity of the rail vehicle and in the ideal case to ensure constant approach time provided that velocity of the rail vehicle in the approach section remains constant. If condition of constant approach time is to be fulfilled, for different speeds of rail vehicles we need different lengths of approach sections. To implement these principles in the model, some conversion rate has to be defined to recalculate real time to internal simulation time-units. Then warning delay and length of virtual approach sections for particular train speeds can be calculated. Obtained results applied in the model are shown in Table 1. Other future settings that are only planned so far are mentioned within conclusions and future work.

Table 1. Delays of warning activation

Train speed [km.h^{-1}]	Delay [s]		Length of approach section [m]
	Number of time-units	Simulation time [s]	
160	0	0	1360,128
150	4,733333	2,024352	1275,12
140	10,14286	4,337897	1190,112
130	16,38462	7,007372	1105,104
120	23,66667	10,12176	1020,096
110	32,27273	13,8024	935,088
100	42,6	18,21917	850,08
90	55,22222	23,61744	765,072
80	71	30,36528	680,064
70	91,28571	39,04107	595,056
60	118,3333	50,6088	510,048
50	156,2	66,80362	425,04
40	213	91,09584	340,032
30	307,6667	131,5829	255,024
20	497	212,557	170,016
10	1065	455,4792	85,008

2.4. Agent types and their communication

The model works with three kinds of agents – patches, turtles and the observer. Patches are stationary and arranged in a grid (exampled in Figure 3). Turtles move over that grid. The observer oversees everything that's going on and does whatever the turtles and patches can't do for themselves. All three types of agents can run NetLogo commands and procedures. A procedure combines a series of NetLogo commands into a single new command that we define. Thus the "level crossing agents" are responsible for controlling the flow of "road vehicle agents" moving safely through the level crossing. They perform actions (operate the signalling lights and barriers). "Road vehicle agents" include the obstacle-avoiding and car-following behaviour. In future another physical attributes (besides colour and type) such as length, acceleration as well as the driver's characteristics (e.g. aggressiveness) could be defined if necessary. The communication between agents is considered to be act/sensing behaviour, anonymous and using event-based interactions. A communication model between agents at the level crossing with 3 in-entries and 3 out-entries is presented in Figure 5 (the used notation was inspired by [13]).

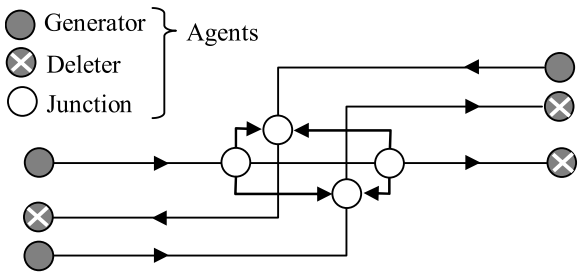

Figure 5. Communication between agents

All messages are issued as actions and received as perceptions. The generator-agent produces messages that are vehicles/information about vehicles. An agent of type junction sends a received message to another agents acting as barriers and signalling. When a message comes to an agent of type deleter, it will be eliminated from the system and will not be considered any more. Basic rules are considered a part of the environment structure and dynamics.

2.5. Implementation

Current implementation of the level crossing system provides a good foundation for further improvement. Implementing the MAS architecture makes it possible to easily add new agents to simulate operation of other really implemented technologies, including ITS technologies bringing supporting functionalities, without affecting operability of other agents. The various modules of the source-code can be reused.

The presented model is able to provide different simulation outputs, available in the bottom part of the interface window:

- Numbers of vehicles in each direction, classified to cars, trucks and a re-calculated number of normalised "unit vehicles" (actually found in the world as well as passed since the start of simulation);
- Road traffic volumes (intensities) observed in particular hours, shown in the form of histograms (based on both measured and normalized data);
- Total duration of simulation running (expressed in virtual days and hours based on defined time-units) and within the actual simulation day (expressed in virtual hours and minutes);
- Trains that passed per day (separately shown for passenger trains, freight trains and all trains together). This parameter is necessary for calculation of the traffic moment (simplified to a product of rail traffic intensity and road traffic intensity recorded in a rush hour); its value can be obtained when a simulation day is over. Example of a typical user message with traffic moment value is shown in Figure 5. This parameter is used by many railway operators when making decision on a choice of level crossing equipment (e.g. level crossings operated by Czech railways must be secured if value of traffic moment > 10^5).

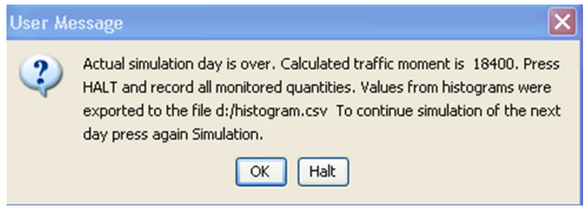

Figure 5. A typical user message containing a calculated value of the traffic moment based of one simulation day

Small sub-window in the right-hand bottom corner of the simulation world is indicating occupancy and/or vacancy of particular track sections. Occupancy of approach section is displayed differently for both possible settings of the *ATPr* chooser: if it is off, the entire approach section becomes coloured in red to indicate occupancy. If it is on, red pointer shows a position corresponding to beginning point of the virtual approach section calculated on the base of actual train velocity (as depicted in Figure 1).

The model works with normalized "unit vehicles": a car corresponds to one vehicle; a caravan, truck, tractor or any other heavy mechanism is equivalent to two unit vehicles, and a motorcycle to 0.5 unit vehicle.

If simulation of the model is terminated in a standard way, statistic data is automatically exported from the model and saved to a separate file for the further analysis.

3. Conclusions and future work

The paper summarizes results of research work that has been concentrated on design of the MAS model of the level crossing system. The model makes possible to test applicability and efficiency of the multi-agent approach and to prove its significant potential for future potential work. There are still a lot of ideas how the model could be further improved:

- Railway vehicles (trains) in the model are generated on the base of pre-defined *Train Graphic Timetable* (TGT) incorporated in the source code or based on user action. For future it would be desirable to define TGT in the form of an external data file, which could be automatically imported into the model during initialization phase. By analogy, similar approach could be applied when generating road vehicles (with a choice of different probability distributions);

- Another extensions could be associated with definition of different track configurations (single-line, double-line or triple-line; one-way or both-way operation) and country-specific interface - graphic interface could be adapted to conventions and different signalling rules applied by different railway operators (different signalling, different operation principles etc.);
- In addition to three actually implemented possible states of the level crossing system (basic, warning, annulation) the model could be extended for operation under faulty conditions. At present it disallows any failure occurrence and/or maintenance simulation;
- To identify and implement other functions (e.g. road vehicle detection) existing in the world and bring this (rather conservative) technology closer to more intelligent and advanced solutions.

In the near future the author intends to publish the entire code of the model within the group of NetLogo community models, which is open to public and available at the www-pages [10].

4. Acknowledgments

This work was supported in part by a grant given to the international scientific-technical co-operation project (MVTS) 6RP/SELCAT "Appraisal and technology of safer European level crossings", and in part by the VEGA grant No. 1/0040/08 "Mathematic-graphical modelling of safety attributes of safety-critical control systems".

5. References

[1] Braubach, L., Pokahr, A., Krempels, K. H., Winfried Lamersdorf, W., "Deployment of Distributed Multi-Agent Systems", *Fifth International Workshop on Engineering Societies in the Agent World*, Eds., Marie-Pierre Gleizes and Andrea Omicini and Franco Zambonelli, 2004.

[2] Tetsuya Yoshida, "Cooperation learning in Multi-Agent Systems with annotation and reward", *International Journal of Knowledge-based and Intelligent Engineering Systems*, No. 11, IOS Press, pp. 19-34, 2007.

[3] France, J., Ghorbani, A. A., "A Multiagent System for Optimizing Urban Traffic", *Proc. of the IEEE/WIC International Conference on Intelligent Agent Technology IAT 2003*, IEEE Computer Society, pp. 411-414, 13-16 Oct. 2003, http://www.bmf.hu/conferences/saci2005/Cico.pdf.

[4] Balbo, F., Pinson, S., "Dynamic modelling of a disturbance in a multi-agent system for traffic regulation", *Decision Support Systems*, 41, pp. 131-146, 2005.

[5] Hirankitti, V., Krohkaew, J., "An Agent Approach for Intelligent Traffic Light Control", *Proc. of First Asia International Conference on Modelling & Simulation,* pp. 496-501, 27-30 March 2007.

[6] Masoud Mohammadian, Multi-Agent Systems for Intelligent Control of Traffic Signals, *Proc. of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce (CIMCA'06)*, p. 270, 2006.

[7] Proença, H., Oliveira, E., "An Adaptive Multi-Agent System for Railway Traffic Control", *GESTS International Transactions*, Vol. 9, No. 1, June 2005.

[8] Janota, A., Rastocny, K., Zahradnik, J., "Multi-agent approach to traffic simulation in NetLogo environment – level crossing model", *Zeszyty Naukowe*, Gliwice, Poland, Zeszyt 59 Transport, No. 1691, pp. 181-188, 2005.

[9] Janota, A., "*Level Crossing Model*", 2005 http://ccl.northwestern.edu/netlogo/models/community/LevelCrossing .

[10] Wilensky, U., "*NetLogo*", Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999, http://ccl.northwestern.edu/netlogo/.

[11] *SELCAT Knowledge Management System* (web portal), 2008, http://www.levelcrossing.net

[12] Poupe, O. et al.: *Zabezpecovaci technika v zeleznicni doprave II*. NADAS Praha, 1990.

[13] Cicortas, A., Somosi, N., "Multi-agent System for Urban Traffic Simulation", *Proc. of the 2nd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence SACI 2005*, Timisoara, Romania, May 12-14, 2005.

A. Janota is a Deputy Head of the Dept. of Control & Information Systems at the University of Zilina (Slovakia). He has 22 years experience in teaching and researching on safety-related control systems, ITSs and AI applications, mostly in transport domain. He received his Ph.D. in Telecommunications from University of Zilina. Since 2005 he has been a member of the Polish Academy of Sciences, Transport Commission in Katowice, Poland.

- *Received date: February 14, 2008*
- *Received in revised form: May 28, 2008*
- *Accepted date: July 2, 2008*
- *Editor: Fumitaka Kurauchi*