Multiagent System-Based Modeling of Traffic Control System Behavior

Aleš JANOTA
Department of Control and Information Systems, University of Žilina
Univerzitná 8215/1, Žilina, SK-010 26, Slovakia

Juraj SPALEK
Department of Control and Information Systems, University of Žilina
Univerzitná 8215/1, Žilina, SK-010 26, Slovakia

Jana ŠEBEŇOVÁ
Department of Control and Information Systems, University of Žilina
Univerzitná 8215/1, Žilina, SK-010 26, Slovakia

ABSTRACT

The paper describes how multiagent systems can be used to describe functionality of road traffic control systems. The introductory part contains a brief survey of actually available traffic models implemented in the hybrid compiler/interpreter NetLogo. These models have been studied and used for design of module-like sample codes. Consequently, they have served as basic building blocks when describing and observing functional behavior of complex traffic control systems. Descriptions of functionality are derived from fluid dynamics which helps to observe required characteristics. The model of railroad level crossing operation is discussed as a typical example of the results obtained within the discussed approach. The educational aspects are also emphasized in a context of the specific university study programs.

Keywords: Multiagent, Functional Behavior, Traffic Control, NetLogo

1. INTRODUCTION

A lot of work has been done so far in the field of traffic modeling based on the multiagent approach. The introductory part of this paper contains a brief survey of identified models. The authors have written it with intention to cover the most decisive models from traffic domain that have been used as a starting point and inspired the work presented later. Only those models are targeted that have been developed in the NetLogo environment [14],[19]. Other modeling approaches and traffic modeling techniques – from car-following models, through submicroscopic models, cellular automata, gas kinetic models resulting in macroscopic traffic equations, mesoscopic or hybrid traffic models to queuing models – are not covered by this paper. Usability of the approach described bellow has been practically proved within the education process where a target group were not traffic but control engineers. Therefore the main attention has been paid primarily to the aspect of functional behavior of control systems even when they are applied in traffic domains.

As the fundamental introduction to the problem of NetLogo-based road traffic modeling the Traffic Basic model could be mentioned [16]. It has shown the simple movement of cars on a highway. Each car follows simple rules for deceleration or acceleration depending on the presence of a car ahead. The model helps to demonstrate how traffic jams can form without any external (centralized) cause. A slightly modified version can be found in the Walker Traffic model [10]. The purpose of this modification has been to study how an idealized model of pedestrians that try to pass the street would work in the setting. More sophisticated two-lane version of the basic model has been available in [18], providing drivers a new option; they can react by changing lanes.

Intersections are another aspect of traffic control that are particularly compelling multiagent systems. In the Traffic Intersection model cars are traveling through an intersection [17]. The user has the ability to control the frequency of cars coming from each direction, the speed of the cars, and the timing of the light at the traffic intersection. Once the frequency and speed of the cars have been selected, the user should run the simulation and adjust the timing of the traffic light so as to minimize the amount of waiting time of cars traveling through the intersection. Another model, called Dangerous Drivers, demonstrates the flow of cars through a 4-way intersection with traffic lights. Each road approaching the intersection has three types of lanes: left-turn, right-turn, and straight [13]. A more complex example is available in the Traffic Grid model where one may control traffic lights and overall variables, such as the speed limit and the number of cars, in a real-time traffic simulation, and try to develop strategies to improve traffic and to understand the different ways of how to measure the quality of traffic [20]. Improved versions of this model show how traffic lights try to *self-organize* to efficiently manage urban traffic [6], [2]. In another heavily modified version of the Traffic Grid model cars drive around in a city, going to destinations determined at random before and during the simulation. After some time, an emergency occurs on one side of the city and drivers are instructed to leave the city in the other direction. The model focuses on only a portion of the city, represented as a grid of eight *streets* by eight *avenues*. Officials can attempt to assist drivers in the evacuation by changing the signal timings of traffic lights when the emergency occurs [5]. Dresner and Stone [4] have proposed a reservation-based system for alleviating traffic congestion, specifically at intersections. The simulator models traffic at intersections and has three different intersection control policies: overpass, traffic light, and reservation system. Under the assumption that the cars are controlled by agents, the authors have claimed that their reservation-based system can perform two to three hundred times better than traffic lights. Another model has been inspired by a real-life incident at a 3-way T-shaped Indian intersection and shows how cars can be running under conditions of lights malfunctioning [11]. Hirankitti and Krohkaew [7] have adopted the approach when each of agents controls all traffic lights at a road junction by an observe-think-act cycle. That is, the agent repeatedly observes the current traffic condition at the junction, it then uses this information to reason with condition-action rules to determine how the agent should act in what traffic condition, and finally it performs those actions in order to efficiently manage the traffic flows. The results obtained through the NetLogo-based traffic simulator can reduce the average delayed time of each car at each traffic-light near a junction rather substantially when compared with other approaches. Similarly, other models can be found dealing with analogical problems, e.g. pedestrian movement [3], [12].

The models discussed above have been studied and used for creation of source code modules representing basic building blocks for more complex traffic control systems whose functional behavior and characteristics have been observed within modeled environment.

2. MAS-BASED APPROACH

Motivation

In this context the multiagent approach is understood as a supplemental approach to modeling complex systems. The NetLogo environment has been used to create a model itself and the environment in which functional behavior of the control system can be simulated and studied. The target group used to evaluate the approach was represented by students completing their MSc. degree program in Control Engineering, i.e. students having previous knowledge on analysis and synthesis of safety-related control systems; dominantly applied in railway and road transport domains. Many of the modeled control systems are deterministic event driven systems whose functionality is normally described with the UML, SysML, Petri Nets, ladder-logic (for PLC-based systems) and/or other traditional formalisms. However, these formalisms usually do not provide possibility to simulate and observe some traffic-based characteristics after they had been put into a certain environment, with certain traffic flows, variable load developing over time etc. Therefore multiagent systems have been applied as a good alternative approach to create:

- A model of an environment as a network or a landscape, i.e. topography with typical static entities such as streets, sidewalks, static obstacles.
- A representation of the traffic flow within the modeled environment where attributes are the geometric forms and sizes of moving entities, and motion obeys physical laws (either macroscopically as flows or microscopically as individual entities [1]).

So, the motivation is to use multiagent systems as a supplementing modeling approach that makes possible to seat the system into certain environment and observe how its behavior is evolving over time. Essential theory background is complemented with some minimum practical experience based on work with the NetLogo environment.

Modeling Process

Models have been created from simple to more complex ones. To make the design process as effective as possible, a database of sample source codes has been built on the base of study of already existing models presented in the introductory survey and newly developed ones. The modeling process itself can principally be seen distributed to several phases (see Figure 1).

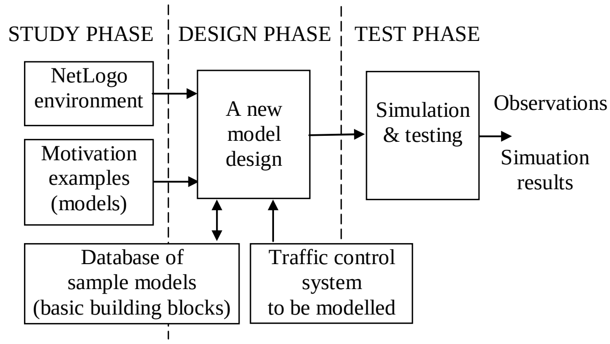

Figure 1 - Modeling process

At first a basic understanding of the NetLogo environment and its abilities is needed, i.e. understanding the agent behavior and its emergent properties. Existing models can be used as a strong motivation factor and/or can be slightly modified to see how different pieces of code affect the functionality of the model and how the software environment can be handled. The NetLogo interface must become managed – how to use tabs, buttons, sliders, switches, monitors and plots, stationary and mobile agents, an observer and links.

Experience obtained with the NetLogo software demonstrates many advantages:

- Simplicity and availability that presents the minimum installation problems.
- User friendly and attractive interface.
- Easy visualizations that attract and increase a modeler‘s interest, help to gain a better understanding of solved problems.
- Well-established user-base and large library of classic and exemplary models that can be tested, experimented and used as a good source of inspiration. Permanently growing number of community models may also be studied, available via Internet from the Center for Connected Learning and Computer-Based Modeling at the Northwestern University.
- Intuitive language that can be easy learned.
- The learning curve seems to be gentle and a decent model may be produced in a short time, receiving immediate feedback.
- Intuitive understanding of fundamental concepts of multiagent systems, no exhaustive and long-term training in computer science is needed.

In the next step simple models can be built piece-by-piece using a pseudo-code approach which involves writing behaviors in comments before using the code. At first attention is paid to stationary agents (called “patches”) organized in a grid and creating the simulation world. A series of subtasks aims to create a surface communication, beginning with one straight lane through intersections to multilane or complex trajectories. Different worlds are inspired by existing models as presented in the introductory survey.

Having basic knowledge on the usage of patches, attention may next be paid to mobile agents (called “turtles”) moving over a grid of stationary agents, i.e. to creating transport means, walkers, static and dynamic equipments etc., from drawing their shapes to applying various principles of their generating, killing, re-directing, moving and/or changing properties. All this is closely connected with explaining setting and monitoring elements available in the NetLogo environment.

Gained knowledge then can be used to create scientific models, i.e. models used for testing hypotheses, analyzing scenarios, drawing conclusions to certain problems etc. In relation to road traffic the following car movement strategies can be realized, creating database of basic building blocks:

- A car is moving at a constant velocity.
- A car is speeding or slowing in response to static side-ways effects (presence of police patrol, a walker standing at a pedestrian crossing, a traffic sign ordering speed restriction).
- Cars are moving with regular spacing at constant velocity.
- Cars are generated with random spacing based on a certain probability distribution.
- Cars react to presence of another car being ahead (rules of decelerating and accelerating).
- Cars are able to change lanes.
- Cars follow certain traffic rules when passing intersections (right of way, bus priority, etc.).
- Cars follow rules given by a certain traffic control system (a traffic light controller, ramp metering, platoons control in automates highway systems, level crossing signaling; etc.).

The last item represents the main target group of models that make possible to solve different problems of optimization, control strategies, traffic operation under working and/or malfunctioning conditions, inevitably based on observing certain parameters of traffic flow or other operation processes. Each agent has some local information and where the goal is to get all the agents to adapt themselves to an optimal state within the system.

Case Study – Railroad Crossing Model

To demonstrate typical outcomes of the present approach the model of a railroad crossing is briefly presented. The model is used to show functional behavior of the studied control/signaling system and obtain some traffic-based characteristics (e.g. traffic moment) reachable by modeling of surrounding environment effects (traffic flows). The very first idea of making the model appeared in 2005 when a concept model was designed and presented in [8]. Then it was used as a motivation example in the AI-related courses. It suffered from many simplifications but became valuable for the acquisition of necessary skills and for getting the modeling environment under control. The new updated model was made more realistic and helped to move attention from *how to do something* to *what/why to do so*. One year later the model was further improved, translated from Slovak to English and presented to the NetLogo community. Technical and implementation details of the model addressing community of intelligent transport system professionals were introduced in [9]. The model itself was uploaded in Jul 2008 to the list of NetLogo community models (http://ccl.northwestern.edu/netlogo/models/community) under the name "Level Crossing ver2_1" where it is publicly available. Since it may be easily run on-line and its quality tested if necessary, no "deep" descriptive details are included in this paper. Control elements applied in the NetLogo interface make possible to set parameters such as road vehicle types, the condition of their arrivals (spacing) and the maximum speed settable separately for each direction, configuration of freight and passenger trains, selection of train graphic timetable, maximum velocity allowed at the railroad line, possibility to generate an auxiliary train and usage of an approaching time predictor. This makes possible to consider and understand causalities and background theory as used in the real life. Functional behavior of the control logic of the railroad level crossing may be defined by a set of implemented functions. Generally, they may be symmetrically divided to the road and railroad parts:

- Detection of railroad and road vehicles.
- Informing of railroad and road users.
- Warning of railroad and road users.
- Protection of railroad and road users.

A relatively detailed description of the mentioned functions is given below. Its purpose is to show wide variety of partial problems and options that could be potentially implemented and explained in the process of creating the multiagent model and its testing.

Detection of railroad vehicles: In the real life railroad vehicles are mostly detected by track circuits or axle counters. To fulfill requirements of the control logic the railroad line is divided into three separate track sections whose logical states (vacancy or occupancy) are monitored. The change of any track section state caused by railroad vehicle movement represents an event that gives the control system a character of the event-driven system. That means a *bottom-up* approach has been applied: the individual parts (lower levels of the simulation) are defined and the overall behavior of the simulation model comes forth from the interaction between the individual parts. In this case, activation of the warning state is derived from detection of a railroad vehicle in the approach section. The basic safety principle applied is saying that the warning state must be activated in such a way that the time between the start of the warning period and the moment when the rail vehicle face is coming to the level crossing must be so long that the longest and slowest road vehicle, just entering a danger zone at the moment of warning state activation, must be able to leave the danger zone safely. According to the national (Slovak) legislation the longest and slowest vehicle is 22 m long, moving at the speed 5 $km \cdot h^{-1}$. It means that approaching times must be tailored to speed of the fastest railroad vehicles, i.e. to the maximum rail line speed. The model enables setting various train speeds. If the place of detection is the same for all railroad vehicles, the negative consequence of slower train existence is that road traffic participants must wait for their coming longer than it would be necessary. Therefore one of the implemented optional functions is a function imitating operation of the approaching time predictor. This equipment, if used, ensures calculation of the speed-dependent point of activation (a fictitious detection point moving inside the approach section) to keep a constant approaching time. Generally, the faster a rail vehicle is moving, the earlier the warning state becomes activated, i.e. detection occurs at a greater distance from the level crossing.

Detection of Road Vehicles: Some level crossing installations may also perform the function of road vehicle detection. Implementation of this function may be motivated by the need to detect obstacles potentially found in the danger zone (a car whose engine failed, the car of an attempted suicide, a swerving driver moving across the rails or any unknown object including pedestrians, cyclists and so on). The model in question does not have this function implemented.

Informing and Warning of Railroad Users: A crew of the rail vehicle may be informed about the status of the level crossing system directly through the rail signal, or indirectly via information obtained from a cab signaling system. The former choice is used in the model. In operation of Slovak railways this signal, if installed, consists of two yellow permanent lights or yellow reflectors arousing train crew's notice and one white line representing status of the level crossing system. If the control system or its monitored action elements (warning lights, barriers, etc.) are working properly the white light is on, otherwise it will be off.

Informing of Road Users: According to the national legislation any level crossing must be equipped with the St. Andrew's cross. In addition, passive information traffic signs are used to inform road users about the existence of the level crossing ahead. They are situated in distances 80 m, 160 m and 240 m from the boundary of the level crossing and its danger zone. In the model they are represented by static patches and their passing might have effect on decreasing actual traffic speed of road vehicles approaching the level crossing. National specialization of the Slovak signaling rules is an optional existence of the so called *active signaling*. It is represented by white flashing lights saying that no rail vehicle is approaching or departing. Originally, it had the meaning of a safety guarantee given by the railroad operator, but due to later modifications of legislation that meaning has been abandoned. Similar signaling can be found in other Central or Eastern European countries, too. If the white *active* light is flashing, car drivers are expected to respect the higher speed limit 50 $km \cdot h^{-1}$ when crossing the danger zone. Otherwise they should not exceed the speed limit 30 $km \cdot h^{-1}$.

Warning of Road Users: Active level crossing systems have two red alternatively flashing lights accompanied by an audible warning. The audible signal is usually generated by a horn or bell. Generally, other advanced possible solutions of warning road traffic participants are available resulting from implementation of intelligent transportation technologies (automatically sent messages warning of the approaching train and interpreted through the in-car navigation system, radio set etc.); however, these advanced options are out of the scope of the discussed multiagent model.

Protection of Railroad/Road users: Road user protection is generally realized by full- or half-barriers (as in this case), gates or some physical barriers preventing road users from entering danger zones, but enabling leaving them if necessary. Railroad user protection is not implemented in the model. In real life it could be represented by some kind of the automatic system able to apply quick-acting braking of the railroad vehicle in case of emergency. This could happen if an obstacle has been detected in the danger zone or the level crossing system has failed.

Outcomes of the Model

One of the outcomes available in the model simulation is the static parameter M called *traffic moment*. It can be obtained at the end of one virtual (simulation) day. This parameter is important when sufficiency or insufficiency of technical equipment installed at the level crossing is evaluated. According to valid legislation the value of M is officially defined as a product of two traffic densities $M = D_{road} \times D_{rail}$, where D_{road} is road traffic density and D_{rail} is railroad traffic density. Calculation of the traffic moment in the model is slightly simplified (e.g. no pedestrians and bicyclists are considered yet), but its principal idea can be demonstrated In the real life the value of M for level crossings installed at the main lines is usually about $5 \cdot 10^5$. 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 future analysis.

Simplifications in the model: It is apparent that a lot of simplifications have been applied in the presented model. In addition to those relating to traffic moment calculation there are some other features that seem different from operation of the real world technical systems:

- For the sake of simplification only single-line traffic has been considered, with trains moving only in the West-East direction, i.e. from the left to the right side of the simulation window.
- No road vehicle may become blocked between full-barriers.
- As road vehicles only cars and trucks are modeled (no buses, tractors, cyclists, motorcycles, pedestrians, etc.).
- No failure or maintenance of the level crossing system is admitted and simulated.

Future possible improvements: The first challenge concerns the visual appearance of the model interface, despite the fact that in the context of multiagent systems and theory hidden in the background it may seem worthless. The actual model interface corresponds to Slovak-like conventions. Not much effort and work has been done so far to harmonize road-railroad interface over Europe or the rest of world. As a negative consequence road drivers may meet totally different signaling systems with potentially misunderstood meanings which may have negative consequence on their safety. Therefore several important research projects have been performed recently with the aim to identify equalities and differences as a necessary condition of future harmonization. The model in question seems to have potential to help in the mutual understanding (could the model be designed with optionally settable interfaces and functions corresponding to more different railroad/road operators?).

Other future improvements could increase variability and eligibility of the model. The main ideas of how the model could be further improved are as follows:

- Single-line track configuration could be changed to double- or triple-configuration with both-way operation.
- Besides expected functioning also operation under failure conditions of the level crossing system could be considered, operating instructions under maintenance activities could also be included.
- The upgraded model could contain other functions existing in the real world and not implemented yet.
- Railroad vehicles (in this case passenger and freight trains) have been generated based on the pre-defined train graphic timetable which was directly entered in the source code. In the future it would be desirable to download the timetable data from external data file during the setup phase of model testing.
- Similarly, more attention could be paid to the probabilistic generation of road vehicles, reaching wider variety of traffic means and higher correspondence to more realistic calculation of traffic moment and other statistic parameters.
- Some specific road and railroad configurations could be simulated to solve specific traffic problems, e.g. so called a blocking back problem arising when road vehicles become blocked inside a dangerous area of the railroad level

crossing due to traffic jam. Understanding traffic dynamics in this or similar cases can not only help to identify reasons for bottleneck, it also contributes to the development of modern traffic assistance systems aiming at the improvement of safety, capacity and comfort.

3. CONCLUSIONS

The paper has outlined how multiagent systems have been used as a supplementary approach to simulation and study of complex traffic control systems. As far as the educational aspects of the presented approach are concerned, it is clear that a hands-on approach is best for helping students to understand dynamics and distributed algorithms and masters this approach to modeling functional behavior of complex systems. The railroad level crossing model is discussed to demonstrate a typical output of the applied approach described above, with a minimum theory of collective intelligence hidden in the background. On the present a library of sample code patterns has been created that would serve as a set of building modules enabling more efficient and faster development of traffic-oriented multiagent applications based on modular approach. A number of partial examples from the simplest to more complex and sophisticated ones seem to be very helpful. The existence and use of such a library enables an increased level of complexity of experimented models within a limited time space.

4. ACKNOWLEDGEMENTS

The paper was prepared under partial support of the Slovak Grant agency VEGA, grant No. 1/0023/08 *Theoretical apparatus for risk analysis and risk evaluation of transport telematic systems,*
and the partial support of the ERDF - European Regional Development Fund, project No. ITMS-26220120028, *"Centre of excellence for systems and services of intelligent transport"*.

5. REFERENCES

[1] A. L. C. Bazzan, F. Klügl, **Multi-agent systems for traffic and transportation engineering**. Bazzan, A.L.C. & Klügl, F. (Eds.), Hersey, PA: Information Science Reference, 446, 2009.

[2] J. C. Burguillo, P. S. Rodriguez, E. Costa, F. Gil, "History-based self-organizing traffic lights", **Computing and Informatics**, 28(2), 2009, pp. 157-168.

[3] J. Dijkstra, H. J. P. Timmermans, A. J. Jessurun, "A multi-agent cellular automata system for visualizing simulated pedestrian activity", **Theoretical and Practical Issues on Cellular Automata - Proceedings on the 4th International Conference on Cellular Automata for Research and Industry**, S. Bandini & T. Worsch (eds.), Springer Verlag, held at Karlsruhe, Germany: October 4-6, 2000, pp. 29-36.

[4] K. Dresner, P. Stone, "Multiagent traffic management: A reservation-based intersection control mechanism", **Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems AAMAS'04**, Vol. 2, New York, IEEE Computer Society, July 19-23, 2004, pp. 530-537.

[5] M. Durak, E. Goodman, **Optimizing traffic signal control for emergency evacuation**. AB-2031, Rev. 07.09, Michigan State University, East Lansing, MI 48823, 2009.

[6] C. Gershenson, **Self-organizing traffic lights**, 2005. Retrieved from http://uk.arxiv.org/abs/nlin/0411066v2

[7] V. Hirankitti, J. Krohkaew, "An agent approach for intelligent traffic-light control", **First Asia International Conference on Modelling & Simulation AMS'07**, doi=10.1109/AMS.2007.11, pp. 496-501.

[8] A. Janota, K. Rastocny, J. Zahradnik, "Multi-agent approach to traffic simulation in NetLogo environment - level crossing model", **Zeszyty Naukowe**, 59(1691), 2005, pp. 181-188.

[9] A. Janota, "MAS model of the level crossing", **International Journal of ITS Research**, 6(2), 2008., pp. 111-116.

[10] S. Kampakis, **The walker traffic model**, 2009. Retrieved from http://encefalus.com/walker_traffic.html

[11] S. Kumar, S. Mitra, "Self-organizing traffic at a malfunctioning intersection", **Journal of Artificial Societies and Social Simulation**, 9(4) 3, 2006. Retrieved from http://jasss.soc.surrey.ac.uk/9/4/3.html

[12] S. Lassarre, A. Godara, (n. d.), **Pedestrian accident risk assessment in urban systems using virtual laboratory SAMU**. Retrieved from http://www.tsc.berkeley.edu/newsletter/fall2007/C3/994/Accident_Risk_Analysis_Paper_new.doc

[13] C. Popa, R. Sanchez, J. Adkins, **Dangerous drivers**, New Mexico, Superconducting Challenge, 2007. Retrieved from http://www.projectguts.org/files/Dangerous%20Drivers.nlogo

[14] S. Tisue, U, Wilensky, NetLogo: "Design and implementation of a multi-agent modeling environment", **Proceedings of Agent 2004 Conference**, Chicago, 2004, pp. 7-9.

[15] J. M. Vidal, **Fundamentals of multiagent systems with NetLogo examples**, 2009. Retrieved from http://multiagent.com/2009/03/fundamentals-of-multiagent-systems.html

[16] U. Wilensky, **NetLogo traffic basic model**, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1997. Retrieved from http://ccl.northwestern.edu/netlogo/models/TrafficBasic

[17] U. Wilensky, **NetLogo traffic intersection model**, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. 1998. Retrieved from http://ccl.northwestern.edu/netlogo/models/TrafficIntersection

[18] U. Wilensky, **NetLogo traffic 2 lanes model**, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1998. Retrieved from http://ccl.northwestern.edu/netlogo/models/Traffic2Lanes

[19] U. Wilensky, **NetLogo**, Center for Connected Learning and Computer-Based Modelling, Nortwestern University, Evanston, IL, 1999. Retrieved from http://ccl.northwestern.edu/netlogo/

[20] U. Wilensky, **NetLogo traffic grid model**, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 2003. Retrieved from http://ccl.northwestern.edu/netlogo/models/TrafficGrid