

AUTOMATION OF EXPERIMENTING WITH VARIOUS INPUT DATA TRAFFIC MODELS

Jan TOŤ

Department of Computer Science and Engineering, Faculty of Applied Sciences,
University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic, tel.: +420 377 63 2483, e-mail: jtot@kiv.zcu.cz

ABSTRACT

Traffic control is one of the current key issues to increase road safety and to minimize congestion and pollution in urban areas. To achieve optimized traffic control, new traffic models are developed featuring algorithms profiting of different input data for particular city areas. Current advanced traffic simulating systems and modelling tools are based on various traffic models, but they have limited possibilities for modification or replacement of the in-built traffic models and corresponding data inputs. Development of new traffic models encompasses repetitive composition of experiments from different data inputs. It is difficult to create an experiment if larger amount of data has to be input manually into the model. Our object is to create an open environment in which traffic models could be connected to various data inputs and experiments with different settings could be launched and evaluated. To simplify the development process, a system prototype based on Extensible Stylesheet Language configurations is proposed to automate the compositions of experiments. The prototype is tested on a particular traffic control model developed at the Institution of Information Theory and Automation of the Academy of Sciences of the Czech Republic. The prototype allows for testing the model on larger areas (North of Zličín in Prague, Czech Republic) and due to the automation, the duration of the experiment compositions was reduced to seconds.

Keywords: traffic modelling tools, traffic models, simulation system, configurations, experiments

1. INTRODUCTION

Computer simulation is a method, which, due to the increasing computational performance, has influenced many areas. The increasing number of vehicles, and the associated traffic problems, gave rise to research in road traffic simulation in order to improve on traffic control. One of the current key issues is to increase safety on roads, to minimize congestions and, especially in urban areas, to minimize the negative impact of emissions and noise produced in road transport, see [1]. Hence many different traffic models were developed and many others are still being developed, especially to cover particularities dominant in complex city areas.

There are many computerized analytical tools – advanced traffic modelling tools (ATMT). They integrate transport modelling software for wide professional applications common in commercial sphere [2]. These tools are very complex and sophisticated systems and they provide user-friendly environment created exactly to fit requirements of the in-built modelling algorithms making them relatively easy to reapply the simulations with different input settings for various situations. Such tools are e.g. Aimsun [4], [5], Paramics [4], Vissim [4], JUTS [6], etc.

In a case, when a new model is developed, one of the above mentioned or similar tools can be used. As illustrated in Fig. 1, such a tool consists of two parts. One part represents one or more “in-built general models” – core modelling algorithms (CMA) – and the second part stands for all services of environment and utilities (SEU). The SEU allows for processing input data set (IDS) within the CMA and it provides user-friendly support to the developer (user/experimenter) during the developing or modelling process. Generally, the development process encompasses network modelling (e.g. network analysis and particular network elements definition and characteristic), traffic modelling (e.g. traffic flows and control plan

programming) and model adjusting (calibration and verification), see [1]. ATMT usually provide automations of most of the parts of the development process and thus it is possible to use them to model large simulation areas in a relatively short time [3].

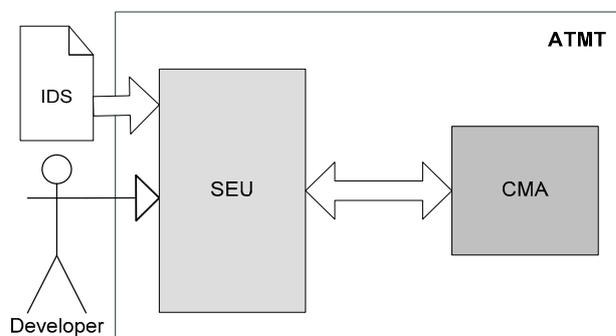


Fig. 1 Schema of a general ATMT consisting of CMA and SEU, where SEU stands in-between the CMA, IDS and the developer

In general, ATMT with in-built CMA and rich SEU guarantee rapid model development, open architecture, API and other interfaces for connection with additional systems. Nevertheless, these tools are limited in following cases:

- The experimental model development requires modification of CMA. (ATMT may have various options, but they are in-built and cannot be modified in any other way than they permit in advance.)
- Modelling process consists of a comparative study among various CMA. (The in-built CMA cannot be replaced.)
- Data required for the modelling are different – IDS origins from various resources, its incomplete or in diverse formats. (ATMT require a predefined IDS in particular format suitable for the CMA.)

It's obvious that developing new traffic models that would not be restricted in above mentioned cases cannot be performed in conventional ATMT. This means that the user-friendly services of SEU cannot be used and the development process requires additional effort to either create new SEU that could cooperate with the CMA (very complex and difficult task, see [5], [6]), or pass through the development process without support services of the environment.

According to [7], the computational complexity of simulation model calibration is NP-complete. When a traffic model development is in the adjusting phase, verification and calibration are required in order to tune the model parameters. The more complex the model is, the more experiments – which run simulations with different IDS, settings or even different CMA – are required.

2. MODEL DEVELOPMENT EXPERIMENTS

Let us say that the first experiment E_1 , performed during the development process of the model, corresponds to an experiment composition (including simulation run) r_1 in order to IDS I_1 is processed within the CMA A_1 to get results R_1 . An experiment E_1 can be expressed as

$$E_1: \{I_1, A_1\} \xrightarrow{r_1} R_1. \quad (1)$$

Developer makes decision D_1 within the process d_1 on the basis of result R_1 about settings of second experiment E_2 , which is needed to be performed with a new set of data IDS I_2 (modification of I_1) within the CMA A_2 (modification of A_1). The decision D_1 is

$$D_1: R_1 \xrightarrow{d_1} \{I_2, A_2\} \quad (2)$$

and the second experiment E_2 based on the result of decision D_1 is

$$E_2: \{I_2, A_2\} \xrightarrow{r_2} R_2 \quad (3)$$

where r_2 and R_2 are analogically the second simulation run, respectively result of second experiment.

If the development process is simplified to a performance of n experiments, it is possible to claim, analogically to previous statements (1), (2) and (3), that the development process P can be expressed as a series

$$P: \{I_1, A_1\} \xrightarrow{r_1} R_1 \xrightarrow{d_1} \{I_2, A_2\} \xrightarrow{r_2} R_2 \dots R_{n-1} \xrightarrow{d_{n-1}} \{I_n, A_n\} \xrightarrow{r_n} R_n. \quad (4)$$

The part $\{I_n, A_n\}$ presents the final settings of the model, and R_n is the result the corresponding experiment provides. In another words, the development process can be understand as a series of experiments compositions and developer's decisions

$$P': \{r_1, d_1, r_2, d_2, \dots, r_n, d_n\}, \quad (5)$$

where the series

$$R': \{r_1, r_2, \dots, r_n\} \quad (6)$$

represents all the experiment compositions (including all the simulation runs) and the series

$$D': \{d_1, d_2, \dots, d_n\} \quad (7)$$

represents all the model developer's decision making processes.

If the IDS I_1, I_2, \dots, I_n of the traffic model are entered manually to the corresponding CMA A_1, A_2, \dots, A_n and there are no utilities to process the results, the processes in series (5) make the development of the model extremely time-consuming and error-prone, see [10]. The series (4) and (5) indicate that the higher number of experiments n represents, the more significant the issue is. The number n grows with the complexity of the model so the manual development of the model becomes very limited and it doesn't allow for developing models manually with larger IDS.

3. OPEN TRAFFIC SIMULATION ENVIRONMENT APPROACH

Our proposal is an approach based on an open traffic simulation environment (OTSE) that serves to develop new traffic models that cannot be raised in the current ATMT. Furthermore, as well as the ATMT, this environment represents a system, that attempts to simplify the development process and to avoid the problems of time-consuming and error-prone repetitive manual actions. Developer's intervention shouldn't be required during process r_i to get result R_i from IDS I_i and CMA A_i of any experiment E_i where

$$E_i: \{I_i, A_i\} \xrightarrow{r_i} R_i. \quad (8)$$

This could be achieved with the corresponding set of SEU that permits to automatize the actions r_i and to help the user in decision making process (7) and consequently development of models with larger IDS can be possible.

Based on these ideas, an OTSE system is proposed in Fig. 2 (the Fig. 2 is an extension of the Fig. 1). The system provides similar SEU allowing for IDS to be processed within CMA and to support the model developer during the developing/modelling process. Besides, communication interfaces I1, I2 and I3 between IDS and SEU, and SEU and CMA in both directions, were proposed in the way as it is demonstrated in Fig. 2. The structures of data that are transferred via these interfaces could be configured arbitrarily with corresponding configurations C1, C2 and C3 so that various IDS/CMA could be used within the system. This also means that an adequate C1 is necessary to provide with each different IDS and adequate C2 and C3 are necessary to provide with each different CMA.

In comparison with conventional ATMT, this configurations-based OTSE enables removing, upgrading and swapping both IDS and CMA. Because of this, the limits of conventional ATMT could be crossed. OTSE permits: model development requiring modification of CMA; modelling using results from different algorithms; using data including inputs from various resources, incomplete data sets or inputs in diverse formats.

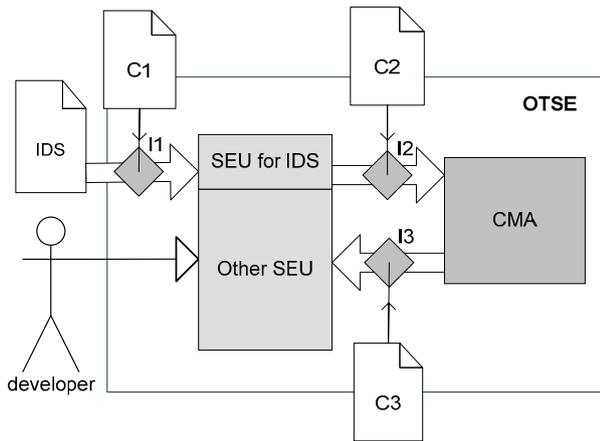


Fig. 2 Schema of proposed configurations-based OTSE approach

3.1. Prototype implementation

The proposed OTSE was implemented as a standalone Java desktop application built on XML and XML-related technologies. The configurations C1, C2 and C3 are sets of Extensible Stylesheet Language (XSL) files. SEU of the prototype integrates an open source XSL Transformation (XSLT) processor, which is able to process the XSL files from each configuration. High scalability of XSL allows for writing transformative sheets for wide range of input/output data structures. The system accepts any IDS and any CMA for which the XSL configurations 1, 2 and 3 are prepared – and these are limited only by the XSLT technology. For XSLT, the requirements include:

First, the IDS must be available in open file formats.

Second, the CMA must feature inputs reading and results writing similarly in/to any open file format.

Third, all data files should be easily parsed and processed within the XSLT processor (e.g. XML, TXT or CSV files).

There is an example of a configuration and its possible realization in the Fig. 3 – the C1 and its role among the IDS, the XSLT processor and inner OTSE data formats, which are the destination for the data from the IDS. The XSL files are used sequentially to transform all data from the IDS files of different formats within the XSLT processor to inner OTSE data formats – XML documents (a.xml, b.xml, c.xml,...), which are generally different from the files from the IDS. The example in Fig. 3 emphasizes the beginning of the transformations – the first transformation controlled specifically by 1.xsl in which 1.txt is transformed to a.xml within the XSLT processor. In the Fig. 3, there could be seen arrows of associations among the XSL files and the source files from IDS, which are considered as inputs for each transformation controlled by one XSL file. Each one XSL file can use data from various IDS files (3.xsl uses both 2.csv and 3.xml) and also any transformation sheet could be divided into various XSL files in complex transformation cases (2a.xsl calls 2b.xsl).

There are SEU with user-friendly GUI available to support the developer's decision making processes (7). Generally, the GUI of these SEU is based on Java's SWING library. To make SEU easily extendable, the concept of separation of the description of the GUI from code

is used in the implementation. This is realized with SwiXML library, which allows to define the static GUI of each component separately (in XML descriptors that are parsed at runtime) from the rest of the code responsible for SEU application logic, navigation flow and user event handlers.

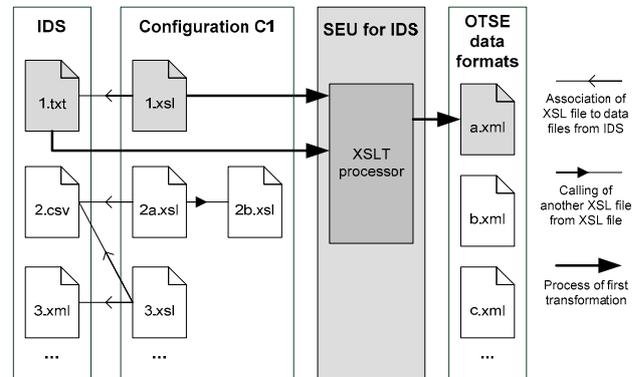


Fig. 3 Example of C1 and its role among IDS, XSLT processor and inner OTSE data format with indication of first transformation process

There are XSLT utilities provided for automation of the composition of experiments in order to make each r_i in (8) run without user intervention. Experiment compositions are automatic one-click set of XSLT transformations, as it is introduced in [10]. The most important SEU utilities implemented in the prototype are following:

3.1.1. Maps Manager

Maps Manager is a tab panel that allows multiple simulation areas to be contained within a single application running instance, which helps the decision process it encompasses experiments on different simulation areas comparison.

3.1.2. Map Viewer

Map Viewer (MV) is a canvas for rendering the selected data of the map inside the application. It's implementation is built on the Apache Batik SVG Toolkit library. The library is responsible for rendering images of Scalable Vector Graphics (SVG). The vector images are obtained as results from XSLT transformations of the XML documents in OTSE data formats. Thus, the MV is able to visualize the simulation area similarly as it was designed in the originating editor.

3.1.3. Properties Viewer

Properties Viewer is a generic tree-view panel with configurable user filters for viewing selected elements from the loaded XML inputs represented in DOM.

3.1.4. Properties Editor

Properties Editor is a utility that allows editing any data from IDS in special user-friendly editors for which the input data types and their possible units are specified. These are used for such values as speed limits in kilome-

tres per hour, capacities in vehicles per hour, choices within enumerations, etc.

3.1.5. CMA integrator

CMA integrator is a utility interconnecting selected CMA with the rest of SEU to make the usage of various CMA transparent to the user. It supports the simulation runs (6) and allows for one click automation of experiment composing and launching based on configurations 2 and 3 (see Fig. 2).

3.1.6. Experiments Manager

Experiments Manager is a utility that serves to explore and manage all available experiments (and their results) which have been performed and automatically stored in the application. It is designed to support the developer's decision making processes (7).

3.1.7. Experiments Results Viewer

Experiments Results Viewer is an evaluating tool used to visualize the results of experiments (figures) and to support developer's decision making processes (7).

3.1.8. TediTransformer

TediTransformer (TT) is a particular utility dedicated to automatic transformations. It transforms maps saved in Aimsun Traffic Editor (TEDI) ASCII formats to XML structure used by the prototype application.

3.1.9. Connections to another systems

Utilities MV and TT were designed to cooperate together to be able to import, view and edit maps from the TEDI. This demonstrates that the prototype of OTSE can cooperate easily with external utilities, even with the editor from conventional ATMT. Because of this, our prototype takes advantages of the TEDI from Aimsun, see [10]. The traffic model development, which would have not been performed in this type of tool due to the limits listed in *Introduction*, may be used if combine with our prototype. In this case, any simulation area can be created in TEDI, then imported to our prototype application in TT and after that viewed in MV and partially edited and used for experimenting with selected CMA together with other data from IDS.

There is a screenshot of GUI of main window together with some SEU of the prototype desktop application in Fig. 4. There are MV (with visible sample intersection), Properties Viewer (panel on the right), Experiments Results Viewer (dialog window with the figure), Experiments Manager (dialog window at the bottom) and bar at the top providing access to CMA integrator, TT and another SEU.

4. CASE STUDY AND RESULTS

The proposed prototype implementation was tested with one of the specialized traffic control simulation model (test model), which is being developed at the

Institution of Information Theory and Automation of the Academy of Sciences of the Czech Republic. In [8], it is explained that the objective of the test model is to simulate the effects of various control plan settings on a selected urban road area. Conventional ATMT cannot be used to support the experimenting with the test model because of the following reasons: Firstly, the development of the model requires modification of CMA, secondly, the data designated for the modelling origin from various resources and are in diverse formats, and thirdly, the model is designated for comparative study among various CMA, see [8], [9]. On this account, it is assumed that this model is fully suitable to be tested as a case study of our prototype.

4.1. IDS

The IDS required by the model encompasses files from various resources. Besides the common inputs – simulation area topology, traffic demand, control plans and simulation global settings – the IDS of the model contains also the data obtained from real traffic data measurements as evolutions of occupancy and intensity collected on road detectors within selected time periods. The formats used in IDS are TXT, CSV, XML document and ASCII Aimsun map files, see [10]. All of these formats are open, so they fulfil the prototype implementation requirements.

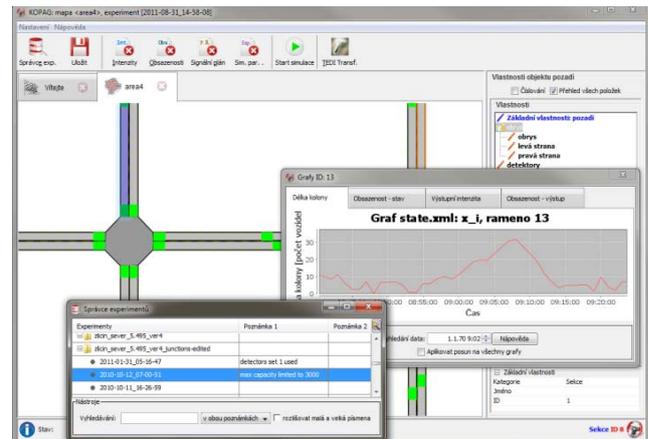


Fig. 4 Screenshot of GUI of SEU of the prototype desktop application with some of the implemented SEU

4.2. CMA

The CMA of the model feature special methods that are experimentally developed to utilize all available information, especially real traffic data measurements. This information is then used to estimate resulting queues lengths on arms of selected intersections after simulations with different traffic control plans are performed. In this case study, there are two versions of CMA available – Matlab application and Java application – which are easily configurable for the OTSE.

4.3. Testing and results

There were three tests performed with the test model in our prototype of OTSE:

4.3.1. Test 1

The objective of the first test was to determine if OTSE is applicable and if it allows for experimenting with the test model. In this test, the test model was used with an area of simple intersection of three arms. This area corresponds to a real intersection in Prague (GPS location is 50° 3' 22.800" N, 14° 17' 13.350" E) and it is visualized in Fig. 5. If the OTSE is not used, it takes from minutes to hours – depending on the changes performed in comparison to previous experiments – to compose a single new experiment manually. First the XSL configurations 1, 2 and 3, according to the IDS and requirements of CMA, were prepared. Then the selected area of intersection map was designed in TEDI. After that, the map was imported using TT into our prototype of OTSE. The result is positive – it is possible to perform the development process (editing, launching, evaluating and managing the simulations experiments in the) using the SEU in our prototype. The left part of the Fig. 5 shows the used simulation area loaded in the prototype application.

4.3.2. Test 2

The objective of this test was to measure minimal experiment composition duration r_i and the duration that a developer spends with a simplest experiments performed in OTSE. The test model with the same simulation area as in the test 1 was used (see the map on the left of Fig. 5). The prototype was launched on an average PC (Intel Core2 Duo CPU P8600, 2.96 GB RAM, Windows 7 x86). Two durations were measured for a set of following synthetic experiments – a set of 23 experiments in that one input parameter was being changed (the speed limit of one road of the test intersection was being increased from 20 kmph to 130 kmph with a step of 5 kmph). Each experiment was edited and launched one by one (moreover, each experiment is automatically saved in order to be later available in Experiments Manager). The decision making process was intentionally skipped (no check of the results of experiments during the test) so that the duration of (7) is zero. Only Properties Editor was used to edit the data before launching each experiment with CMA Integrator. It was measured that during this test, the average duration of composition of such experiment using the selected SEU was 13 sec. (with editing one input parameter) and the duration of r_i (without editing) was 4 sec., see summarization in Table I.

4.3.3. Test 3

The objective of the last test was to repeat the test 2 on a larger area. For that reason a network of intersections in North part of Zličín in Prague was selected. This area encompasses also the intersection used in the tests 1 and 2 – it is the northernmost intersection in the simulation area – see the imported map on the right of Fig. 5. It was measured that during this test, the average duration of composition of such experiment using the selected SEU was 14 sec. (with editing one input parameter) and the duration of r_i (without editing) was 5 sec., see summarization in Table I.

Table 1 Summary of experiments composition duration with the test model using OTSE

Testing area	Average composition duration r_i	
	Including simple editing	Excluding any editing
One intersection ^a	13 sec.	4 sec.
Network of intersections ^b	14 sec.	5 sec.

a. GPS location 50° 3' 22.800" N, 14° 17' 13.350" E, North part of Zličín in Prague, Czech Republic

b. Southern area from the intersection at mentioned GPS location

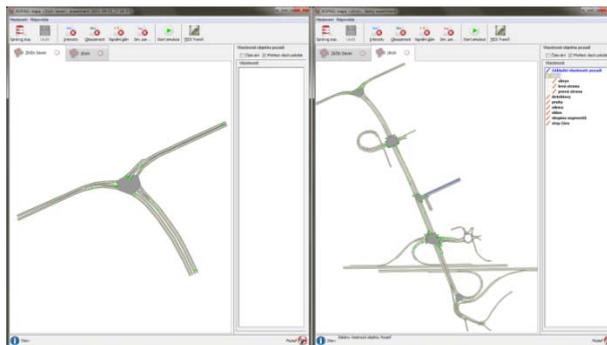


Fig. 5 Visualisation of test simulation areas – a simple intersection (on the left) and a larger area (on the right) – both from the part of Northern Zličín in Prague, Czech Republic

5. CONCLUSION

In this paper, an open traffic simulation environment was proposed. It is suitable for such situations when conventional advanced traffic modelling tools cannot be used, but supportive services and utilities are essential for successful model development.

For this purpose, a configurations-based architecture was introduced. In comparison with conventional advanced traffic modelling tools, such architecture allows for that any core modelling algorithms and their corresponding data input set can be used via the supportive services and utilities. This presented approach was realized as a fully working prototype application, which implements the proposed system in Java and XSL. It features basic services/utilities in order to automate the processing of data inputs, to reduce significantly the amount of information entered manually, and to simplify as much as possible the repetitive work of an experimenter.

The prototype was tested with a real traffic control model and promising results were obtained. The prototype makes possible rapid compositions of experiments (duration of single experiment composition was 4-5 seconds) comparable with standard of conventional modelling tools. This allows for using the test model on larger simulation areas than before.

There are several problems that are going to be part of potential future work in this topic. One of them is to apply performance tests on the prototype with detailed and large simulation areas corresponding to whole cities. Another thought is to extend the proposed architecture to allow online comparative studies of different CMA/IDS to support designers of CMA use.

REFERENCES

- [1] MARTÍNEZ, J. – TOMÁS, V. R. – SAMPER, J. – MARTÍNEZ, J. J. – SÁCHEZ, I. – SORIANO, F.: Simulación de tráfico. Sistemas de control de tráfico. (Traffic simulation. Traffic control systems.), Universidad de Valencia, 2009.
- [2] Federal Highway Administration – United States Department of Transportation: Traffic Analysis Tools by Category, 2004, http://ops.fhwa.dot.gov/trafficanalysis/tools/tat_vol2/sectapp_e.htm
- [3] GARCÍA, F. – ESTEFANÍA, C. – AGUIRRE, J. M. – LARRAURI, J. M. – TOMÁS, V. R. – SAMPER, J.: El uso de simuladores como herramienta de apoyo en la gestión del tráfico (The usage of simulators as a supportive tool for traffic control), Departamento de Obras Públicas y Transportes, Universidad de Valencia, 2009.
- [4] HIDAS, P.: A functional evaluation of the AIMSUN, PARAMICS and VISSIM microsimulation models, Road and Transport Research, Vol. 14, pp. 45–59, Dec. 2005.
- [5] BARCELÓ, J. – CODINA, E. – CASAS, J. – FERRER, J. L. – GARCÍA, D.: Microscopic traffic simulation: A tool for the design, analysis and evaluation of intelligent transport systems, *Journal of Intelligent & Robotic Systems*, Vol. 41, pp. 173–203, 2004.
- [6] HARTMAN, D. – HEROUT, P.: JUTS – J-Sim Urban Traffic Simulator, Proceedings of the International *Workshop Modelling and Simulation in Management, Informatics and Control (MOSMIC)*, Žilina, 2003.
- [7] HOFMANN, M.: On the Complexity of Parameter Calibration in Simulation Models, *The Journal of Defense Modelling and Simulation Applications, Methodology, Technology*, Vol. 2, No. 4, pp. 217–226, Oct. 2005.
- [8] KRATOCHVÍLOVÁ, J. – NAGY, I.: Model dopravní mikrooblasti (Traffic model of a microregion), ÚTIA AV ČR and Faculty of Transportation Sciences ČVUT, Prague, 2004.
- [9] NAGY, I. – HOMOLOVÁ, J. – PECHERKOVÁ, P.: Dopravně závislé řízení silničního provozu ve městech, *Automa*, Vol. 13, No. 6, pp. 12–26, Prague, May 2007.
- [10] TOŤ, J.: Application for data preparation of traffic simulation experiments. M.S. thesis, University of West Bohemia, Pilsen, Mar. 2010.

Received January 18, 2012, accepted June 13, 2012

BIOGRAPHY

Jan Tot' was born on 17. 12. 1984. In 2010 he graduated (Ing.) with thesis "Application for data preparation of traffic simulation experiments" at the department of Computers Science and Informatics of the Faculty of Applied Sciences at University of West Bohemia. He plans to defend his PhD in the field of distributed systems in 2014. His scientific research interests include traffic simulation, multiagent systems and intelligent transport systems.