Upgrade of the MovSim for Easy Traffic Network Modification

Andraž Omahen*, Teo Kukuljan†, Vida Vukašinović‡, Peter Korošec‡ and Gregor Papa‡
*Faculty of Computer and Information Science, Ljubljana, Slovenia
Email: andraz.omahen@gmail.com
†Faculty of Mathematics and Physics, Ljubljana, Slovenia
Email: teo.kukuljan@student.fmf.uni-lj.si
‡Computer Systems department, Jožef Stefan Institute, Ljubljana, Slovenia
Email: {vida.vukasinovic, peter.korosec, gregor.papa}@ijs.si

***Abstract*—Microscopic traffic models describe the individual elements of transportation system, such as the motion of individual vehicle. In this paper, MovSim, an open-source tool for model investigation and simulation, is used and upgraded as a comprehensive tool for building microscopic traffic models. The upgraded simulator allows the construction of complex road networks with many junctions and roads with various traffic signalization and regime. In addition, road network editor is implemented in order to allow easy and user friendly road network reconstruction.**

***Keywords*–Traffic simulator; MovSim; microscopic traffic model; road network editor.**

I. Introduction

The reason for growing interest in traffic simulation models lies not only in their potential for modelling traffic processes, but also their ability to visualize the results in a realistic and appealing way [1].

Traffic network simulation is an important approach for modelling the dynamics of the traffic systems [2]. When implemented with enough details of the real network it helps analyzing the locations and causes of congestions. In this manner, it can offer solutions for uninterrupted traffic flows and traffic safety. The way of traffic congestion evaluation significantly affects transport-planning decisions [3].

Traffic jams occur at any hour, either weekdays or weekends. Drivers face extra travel time, extra cost from wasted fuel, lost productivity as well as increasing unreliability where bad weather, roadwork, a malfunctioning traffic signal, a local event, a small accident, or stalled vehicle can result in major delays [4]. In the 2014 Annual Report, INRIX reveals that congestion levels rose in over half (53%) of European cities.

The microscopic-based simulation provides a detailed representation of the traffic process, being suitable for evaluation of complicated traffic facilities and intelligent transportation systems that often consist of complex traffic management, safety and information systems [1]. However, mesoscopic and macroscopic simulation represent the traffic dynamics with less details, but are faster and easier to apply and calibrate. While macroscopic approach is more suitable for modelling large networks, microscopic approach is usually applied to smaller areas.

To allow detailed and accurate traffic simulation, including congestion evaluation, the road network needs to be described with sufficient details. These include sections lengths, number of lanes, curvature, lane restrictions, etc. All details have a great influence on simulation results. In this paper, we present the upgraded Multi-model open-source vehicular-traffic Simulator, MovSim, which supports complex road network construction. The implemented GUI allows easy road network adaptation to the appearing network topology changes. Even when all details form the real-world are captured in a simulation model, the upgrades and updates of the road network characteristics are still allowed. This includes for instance the possibilities to easily close/open some road section, to easily change the traffic light control sequence, as well to easily adapt the topology of the complex road crossing.

The rest of this paper is organized as follows: Section II lists some related work on traffic simulations; Section III briefly presents the simulator that was used as an origin for our upgrades; the upgrades are described in Section IV; road network editor is described in Section V; while Section VI draws the concluding remarks.

II. Related work

Traffic simulation models can be classified into [5]:

- microscopic - approaching individual elements of transportation systems, such as individual vehicle dynamics and individual traveler behavior;
- mesoscopic - approaching transportation elements in small groups, within which elements are considered homogeneous;
- macroscopic - approaching aggregated characteristics of transportation elements, such as aggregated traffic flow dynamics.

MovSim [6] is an interactive Java-based open-source traffic simulator. In contrast to commercial simulators, the focus is on investigating fundamental issues of traffic dynamics rather than simulating specific road networks. Multi-agent Transport Simulation, MATSim [7] provides a framework to implement agent-based transport simulations. The framework consists of several modules which can be combined or used stand-alone. Modules can be replaced by user's own implementations. Simulation of Urban MObilitiy, SUMO [8] is an open source, microscopic and continuous road traffic simulation package designed to handle large road networks. It allows modelling of intermodal traffic systems including road vehicles, public transport and pedestrians. It can be enhanced with custom models and provides various APIs to remotely control the simulation.

On the mesoscopic and macroscopic level of traffic simulation the following models are available. Mesoscopic Traffic Simulator, Mezzo [9] is a discrete-event mesoscopic traffic

 ISBN: 978-1-61208-442-8

simulation model that simulates road traffic on the level of individual vehicles, but with an aggregated behaviour on links. The model is especially designed to simulate large networks and can be used in combination with a microscopic simulator to work as a hybrid model. OmniTRANS [10] is a macroscopic multimodal and multitemporal system, suitable for modelling the interactions between the various means of transport within an urban context. PTV Visum [11] is a macroscopic traffic analysis and forecast tool that provides a comprehensive range of functions for all aspects of transportation planning and engineering.

III. MovSim traffic simulator

The aim of vehicular-traffic simulator MovSim [6] is to be an open-source tool for model investigation and simulation. It can be classified as a microscopic lane-based traffic simulator. It is able to simulate many basic traffic situations. Simulated vehicles are able to react to a traffic light, and are able to overtake. Lane changes are modeled with the general (minimizing overall braking induced by lane change, MOBIL) strategy based on accelerations [12]. Simulator is supported with graphical user interface, where simulated vehicles dynamics is visualized. It is written in Java and Maven is used as management tool.

The simulator offers the implementation of vehicles inflows and outflows. Inflows are points where vehicles enter the simulated traffic network. They are defined by the network position and the numbers of vehicles entering the network per second. Outflows are points where vehicles exit simulated network. The construction of simple single lane road segment with inflow and outflow, where vehicles are moving from left to right with a setting of input flow to 1000 vehicles per hour is shown in Figure 1.

Figure 1. Single-lane road with vehicles flow.

For every vehicle a lane-changing politeness parameter is assigned, which vary the motivation for lane changing from egoistic to more cooperative driving behavior. In this respect solutions used in MovSim, such as the lane-changing MOBIL passing rules and longitudinal car-following model Intelligent Driver Model (IDM) [13] are essential for simulation of multilane traffic.

A code describing the following traffic cases was available:

- roundabout;
- road block;
- inclusion road in traffic;
- speed limits;
- behavior of vehicles driving on ramp.

In order to built real complex traffic network with many junctions and roads with multiple lanes and various traffic regimes, some upgrades of MovSim were needed. In this paper, upgraded simulator with an editor for easy manual road reconstruction of existing network is presented.

IV. Simulator upgrade

A. Simple junction and lanes with multiple sources and sinks

In order to support the simulation of more complex road structures, such as junction the simulator was upgraded with functionality which allows a leap from one to several road segments. This was required to build even a basic single-lane junction. A basic single-lane junction consists of four input and four output road segments, each having one lane. In basic single-lane junction four roads intersect in a way such that the change from one road to another is allowed.

Upon entering a basic single-lane junction there are three options, the vehicle can turn left, right, or go straight. Similarly, vehicle which leaves the junction could come from three directions. The original MovSim supports only one source and one sink lane segment for each lane. Therefore, the simulator was upgraded with necessary changes so that each lane can have more sources and sinks. Further, in the markup language we had to define a junction. A basic junction consists of four input road segments, four output road segments, and 12 road segments that connect input and output road segments. For an illustration see Figure 2.

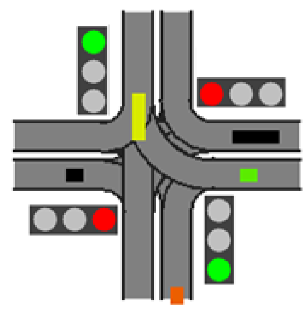

Figure 2. Basic junction.

Then, basic junctions were upgraded by adding traffic lights and setting their lights duration. All four traffic lights are automatically synchronized only by setting the duration of green, yellow, and red of one traffic light direction.

By the simulator upgrade, which allows assigning several sinks to one lane in the road segment, the vehicle route simulation is not trivial anymore. For example, when vehicle comes to the junction, it has no information, which road it should take (left, right, or straight in the basic junction). Therefore, to prevent errors, a sequence of consecutive road segments (path) needs to be defined in advance for every vehicle. In practice, all possible routes from a given source to a given sink are calculated. When a vehicle is generated, one of the possible routes is chosen at random and assigned to it. Based on the roads and their connections, a graph is generated. Its vertices represent links between road segments (sources and sinks of each segment) and edges represent road segments. A graph was build using Java library JGraphT [14]. In order to determine most favorable routes between source and sink (origin and destination) Dijkstra algorithm [15] was used.

B. Critical parts of roads

It seems to be impossible to avoid intersection of road segments in the complex road networks construction. Even in

 ISBN: 978-1-61208-442-8

road network with basic junction such situation appears. A nice example of critical road segments intersection in basic junction is when cars are turning left. A car cannot turn left until the road to be crossed is empty. In order to prevent car crashes, two critical segments, named $road_1$ and $road_2$, of intersecting roads are defined:

- $road_1$ represents problematic part, in which road segments intersect;
- blockade on $road_2$: blockade is set, while $road_1$ is busy (e.g., some vehicle is driving on it).

Traffic through critical parts are also adjusted by defining acceptable speed of vehicles on the critical parts of the road. If the approaching vehicle drives very slowly through the $road_1$, then the blockade on the $road_2$ for turning left is not needed since the turning vehicle has enough time to turn left.

C. Road network construction

Our testing case of real-world road network construction was Ljubljana road network within the Ljubljana ring. MovSim simulator does not have any tool for easy real-world road network construction. Basic small road network construction was done with the help of image of a city map, which was set as the background in the canvas of graphical user interface. City map was not enough to construct the road network, since on the map, it was not evident how many lanes are in input road segments of a junction. Such vagueness was manually solved with the help of Google street-view service. The construction of simple basic junctions were automatized to some level, while the construction of complex junctions is facilitated but it still needs to be done manually.

An example of the detailed construction of small subnetwork consisting of three complex junctions, where Tivolska, Celovška and Slovenska streets of Ljubljana city intersects, can be seen in the Figure 3. Each road segment is identified by unique ID number. Each junction has input roads, output roads, roads within the junction as well there are roads that connect junctions.

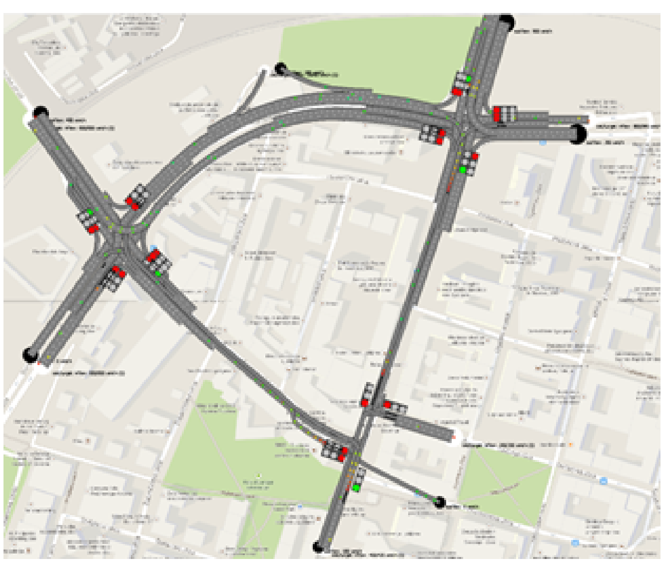

Figure 3. Partial road network.

In the complex network (roads with multiple lanes and multiple output roads), the following problems were detected.

Increased number of vehicle crashes. When larger number of vehicles entered the network, they started to crash. The reason is that the relaxation of vehicles is computationally time consuming. Relaxation is calculation of vehicle priorities order. Most of the collisions occurred during the transition of vehicles from different road segments that have the same sink/target segment. For example, when two vehicles from different road segments try to enter the same road segment at the same time, the calculated positions are the same and the collision is recognized. The problem was solved with similar observers, such as critical parts of the road segments, which were used for simple basic junctions, when vehicles are turning left.

Some vehicles cannot continue their routes. As already mentioned, the route from the source to the sink is assigned to the vehicle before it enters the network. It turns out that some vehicles cannot continue its route. The most common reason is an inability to change the lane. If the vehicle cannot change the lane, it continues its ride to the segment defined as a successor of the current lane. Such vehicle is lost since its road position deviates from originally assigned route. A simple solution can be to simply exclude lost vehicles from the traffic. To make the simulation closer to the real-world situation, the problem was solved by the route recalculation; i.e., a new route from the current road position to the sink is assigned.

D. A network of several roads within the Ljubljana ring

We constructed a network consisting of Ljubljana ring, main basic junctions inside the ring, and roads connecting them (see Figure 4). During this construction the necessity of automatization and facilitation of such network construction arose. Junctions were positioned by mouse clicks and a code of all road segments for all junctions was generated in XML markup language. The result was a set of junctions with missing interconnections. To achieve correct interconnections road identification numbers were needed. Hence, the road network editor for easier and quicker road network composition was required.

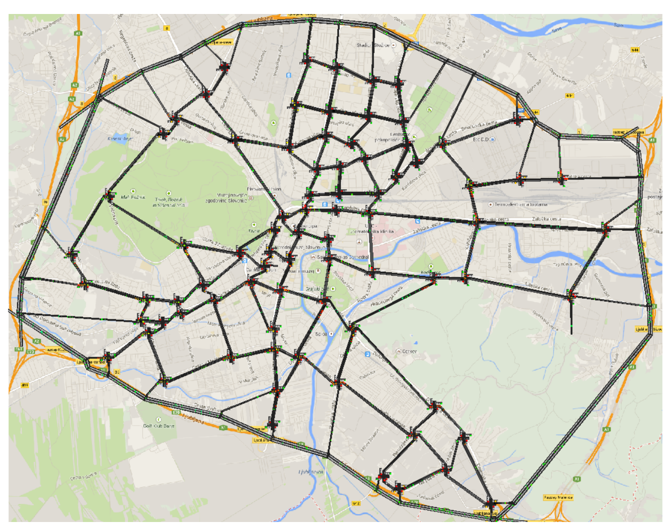

Figure 4. Roads within Ljubljana ring.

We have also started much more demanding part of the road network editor upgrade, i.e., building a parser for automatic road network construction based on OpenStreetMap [16]. The first thing to do was detecting where roads intersect. When positions of all junctions were calculated, it was necessary to get details of each junction from geometry of roads that intersect in that particular junction. Simple directed graph turned out to be very useful for both storing data and converting it to roads and junctions.

V. ROAD NETWORK EDITOR

We built a graphical user interface in order to facilitate the composition of road networks and to avoid manual entering of every individual road segment into the OpenDRIVE [17] file.

Road network editor supports different view options. During the traffic simulation, user can choose the content of the canvas. Possible elements are: map of the city, road segments, traffic lights, ids of roads, sources of vehicles, sinks of vehicles, and speed limits, see Figure 5.

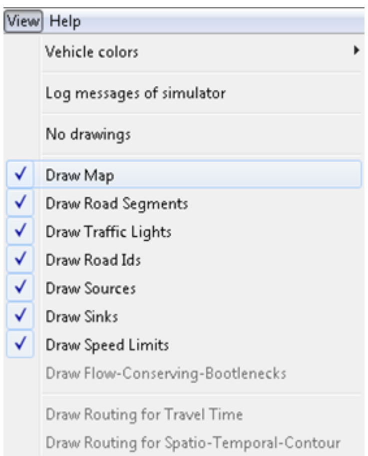

Figure 5. View options.

The most important part of the upgraded GUI is an Edit option which is added to the GUI's toolbar. By selecting edit mode, the simulation stops and the editor is prepared for applying network reconstruction. In edit mode, new road segments and junctions can be created as well as new traffic lights with their light intervals setting can be added. In the edit mode a virtual wrapper around segments is created for easier segment identification and selection.

If straight road is added, the information about x and y coordinates, segment length, and angle under which the road lies are shown in the toolbar. If the curved road is added, also radius and arc length are shown in the toolbar. By pressing the Apply button, the road is added to an existing network but it is not stored in XML structure yet. If we want the road to be stored in the network, we need to explicitly save changes.

Road editor supports adding interconnections of road segments, which are essential for vehicle recognition of the following road segment on the route (see Figure 6). By adding the interconnection of two roads, the lane which represents successor (sink) and lane which represents predecessor (source) must be defined.

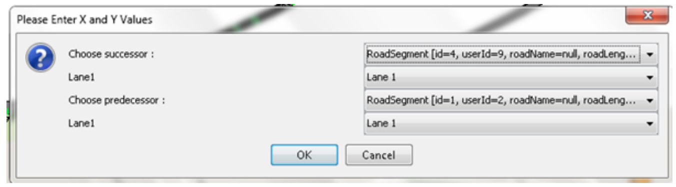

Figure 6. Connection dialog box.

A. Adding junctions

The first problem in a junction construction was to decide which properties of junctions were sufficient for reconstruction. It turned out that there are many properties that need to be determined in order to define junction geometry. These needed properties are evident from the later explanation of junction editor. Further, it is not easy to build a junction out of numbers describing it using just simple geometric shapes as lines and circle arcs. Finally, roads going through the junction need to be linked so that the constructed junction could be used in simulation.

An option for adding simple junction was added in the road editor. By choosing junction scope, Junction option appears. By clicking it (see Figure 7), a Junction editor window opens

Figure 7. Editing toolbar.

(see Figure 8). The user can choose among several junction types, e.g., T, Y, or +. Further, the user needs to define width of roads within the junction, lengths of external roads (incoming and outgoing roads), (x, y) position, rotation angle of the junction, and the number of incoming and outgoing roads for every leg of the junction. For example, if the junction of + type is chosen, then there are 4 legs, and for each of them the number of incoming and outgoing roads must be defined.

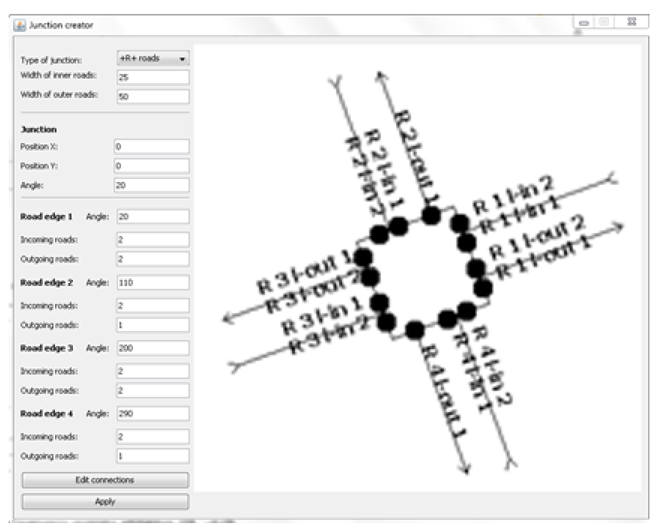

Figure 8. Junction editor.

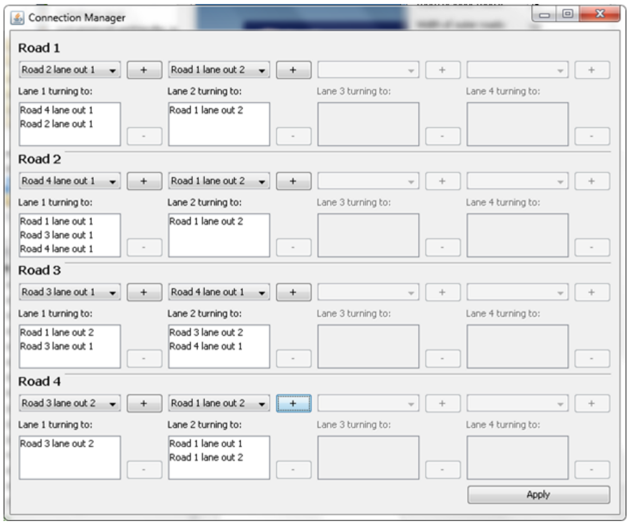

Figure 9. Editing toolbar.

Note that the above-mentioned information is not enough for the junction definition. This information would represent only a set of roads with some dimensional properties. To make it complete, individual roads inside the junction must be connected. Once we fulfill all the information that we want for the junction, an Edit Connections option appears. By clicking it, Connection Manager window opens (see Figure 9). This editor allows connecting incoming and outgoing roads. Each edge of the junction can have up to four incoming roads. For each incoming road the outgoing roads with which it is associated are listed.

B. Traffic lights editor

Traffic lights are mostly assigned to the lanes on the roads in junctions. Different traffic light intervals/phases and their duration in the junction are mutually dependent and are uniquely determined by assigning adequate Controller Group. Although many junctions may have the same traffic light regime, their phases do not always overlap. Mutual dependence of periodic repetition of different light phases in traffic lights in different junctions are controlled by assigning starting time in each junction.

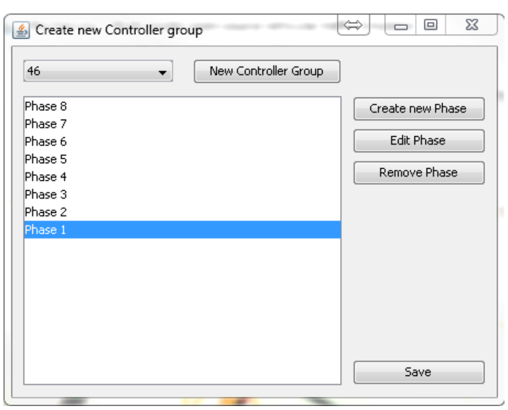

Figure 10. Controller Group editor.

Editing of traffic lights consists of Controller Group part and traffic light location part. First, in the Controller Group part, the light regime for a given set of traffic lights is defined (see Figure 10). The set of traffic lights typically refers to the traffic lights of the junction. Second, the location of traffic lights is determined. These two options can be found as two buttons under traffic lights scope option. The buttons are named Create Group Controller and Locate Traffic Light. In Create Group Controller editor controller groups of traffic lights can be managed. New controller groups can be optionally added, as long as we need them, or they can be selected among the existing ones. Each Controller Group has phases, where each phase is defined by its duration and color of traffic lights. Phases can be added, edited, or removed. Phases are shown in list where each of them can be selected and edited. If we want to edit selected phase, we click Edit

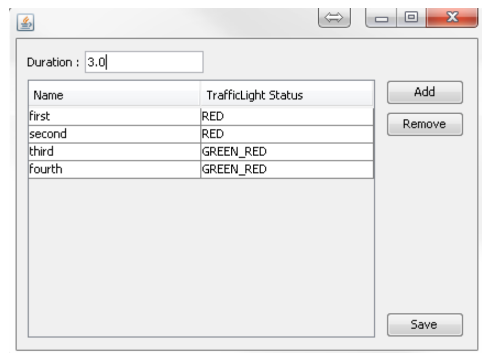

Figure 11. Phases editor.

Phase. In the Phase Editor, we have to specify the length of the phase (see Figure 11). Traffic lights can be optionally added and removed. When the editing of the phase and controller group is finished and saved the data is written to the XML OpenDRIVE structure.

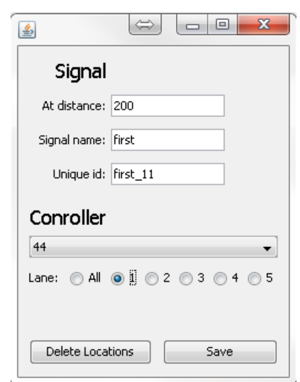

Figure 12. Traffic light locator.

In order to locate traffic lights on the road, the road segment must be chosen. By choosing the road segment, Locate Traffic Light editor opens (see Figure 12). In the editor, the distance from the ending point of the road is specified. Then, the same name as it was set in phases in Controller Group must be entered. Further, an unique id number by which the traffic light is recognized by others is assigned. Finally, the Controller Group to which it belongs to and the lane on which it works is selected.

VI. Conclusion

The focus of this paper was given to the development and implementation of microscopic traffic simulation, which can be helpful at analysis of existing traffic congestions in different city parts. Such microscopic traffic simulation can offer a solution for uninterrupted traffic flows and traffic safety. Since a comprehensive open-source traffic simulator MovSim already supports many options for model investigation and simulation, we upgraded it so that it supports the construction of complex road networks with many junctions and road segments with multiple predecessor and successor road segments. In addition, we built an user-friendly tool for a real-world road network construction in order to facilitate the construction of real-world road networks and to avoid manual entering of individual road network parts into the XML OpenDRIVE file. GUI supports automatic construction of simple junctions with all internal road segments and traffic lights with an option of light phases adjustment.

In the future, we plan to built the parser for automatic road network construction based on the OpenStreetMap data where information will be automatically considered and included.

Acknowledgment

This work was partly funded by the Slovenian Research Agency through the research program ”Computer Structures and Systems”, and was also supported through ARTEMIS-JU funding for the project ”ACCUS – Adaptive Cooperative Control in Urban (sub) Systems” (grant number 333020). [?]

References

[1] W. Burghout, “Hybrid microscopic-mesoscopic traffic simulation”, Doctoral Dissertation, Royal Institute of Technology, Stockholm, Sweden, 2004.

[2] M. Treiber and A. Kesting, “Traffic Flow Dynamics - Data, Models and Simulation”, Heidelberg, Springer, Oct. 2013.

[3] T. Litman, “Comprehensive Evaluation of Traffic Congestion Costs and Congestion Reduction Strategies”, August 2015, URL: http://www.vtpi.org/cong_relief.pdf, [accesses: 2015-09-02].

[4] “INRIX, Urban Mobility Scorecard Annual Report”, URL: http://inrix.com/scorecard/key-findings-us/, [accesses: 2015-09-02].

[5] R. Akçelik, “Traffic models”, URL: http://www.sidrasolutions.com/Resources/Models, [accesses: 2015-06-03].

[6] M. Treiber and A. Kesting, “An Open-Source Microscopic Traffic Simulator”, *Intelligent Transportation Systems Magazine*, vol. 2, no. 3, pp. 6–13, 2010.

[7] “MATSim”, URL: http://www.matsim.org/, [accesses: 2015-06-03].

[8] “SUMO - Simulation of Urban MObility”, URL: http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/, [accesses: 2015-06-03].

[9] “Mezzo - Mesoscopic Traffic Simulator”, URL: http://www.ctr.kth.se/mezzo.php, [accesses: 2015-06-03].

[10] “OmniTRANS”, URL: http://www.dat.nl/en/products/omnitrans/, [accesses: 2015-06-03].

[11] “PTV Visum”, URL: http://vision-traffic.ptvgroup.com/en-us/products/ptv-visum/, [accesses: 2015-06-03].

[12] A. Kesting, M. Treiber, and D. Helbing, “General Lane-Changing Model MOBIL for Car-Following Models”, *Journal of the Transportation Reasearch Board*, no. 1999, pp. 86–94, 2007.

[13] M. Treiber, A. Hennecke, and D. Helbing, “Congested traffic states in empirical observations and microscopic simulations”, *Physical Review E*, vol. 62, no. 2, pp. 1805–1824, 2000.

[14] “JGraphT”, URL: http://jgrapht.org/, [accesses: 2015-07-01].

[15] E. W. Dijkstra, “A note on two problems in connection with graphs”, *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[16] “OpenStreetMap”, URL: https://www.openstreetmap.org/, [accesses: 2015-07-01].

[17] “OpenDRIVE”, URL: http://www.opendrive.org/, [accesses: 2015-07-01].