Open Traffic: a toolbox for traffic research

G. Tamminga\textsuperscript{a,b,*}, P. Knoppers and J.W.C. van Lint\textsuperscript{b †}

\textsuperscript{a} Grontmij Nederland B.V., Postbus 203, 3730 AE De Bilt
\textsuperscript{b} Delft University of Technology, P.O. Box 5048 - 2600 GA Delft, The Netherlands

Abstract

Open Traffic is an open source software project that provides a transport modeling software environment. While most transport model packages offer ready-to-use modules for end-users, Open Traffic provides open access to a modelling environment for the (further) development of methods and algorithms and enables the sharing, distribution and further development of the implied knowledge. The Open Traffic platform is designed as a modular system which enables users to utilize existing modules and extend the system with new ones. The system supports the development of multi modal and multi scale models by providing a collection of objects that enable the creation of a transport infrastructure and its environment at multiple levels of detail. The definition of the geographical objects aligns to the principles of CityGML, an open standard for geo data that is internationally accepted by the Open Geospatial Consortium. Additional utilities such as a graphical editor and visualizer, in addition with facilities to import data from external sources like Open Streetmap and Esri shape files, enable users to quickly create and demonstrate their use-cases. In this article we present the high level architecture of Open Traffic, its current status, and a first application with the implementation of the micro simulation model MOTUS. Also, the possibilities and requirements to adhere Open Traffic to agent based modelling approaches are explored.

© 2014 The Authors. Published by Elsevier B.V.
Selection and peer-review under responsibility of Elhadi M. Shakshuki.

Keywords: transport model; simulation; open source

* Corresponding author. Tel.: +31-30-220-7413.
E-mail address: g.f.tamminga@tudelft.nl
1. Introduction

Transport model packages that are aimed for use in research and science are scarce. While the development of many of the transport model packages actually has been initiated at research institutes, they evolved into products for end-users with the aim to develop transport model applications. User friendliness of these packages is of primary importance, enabling users to create efficient work flows and providing utilities to import, create and edit traffic networks, estimate and predict traffic demand, and simulate and visualize vehicle movements and outputs. For a large share of the model appliers, this provides sufficient functionality.

Due to the fact that most of these packages deploy a closed software approach, the use of the source code is restricted. Researchers who want to investigate the underlying methods and possibly improve, modify or extend them, are unable to do so. Alternatively, a lot of researchers start from scratch and create programs that are dedicated to their specific research questions. In these research projects, the quality of the software, re-usability of the code and possibilities to extend the program for other purposes is not of primary importance. As a consequence, the software will rarely be used by other researchers. To prevent this loss of knowledge and experience, an approach is required where code reuse and proper documentation is stimulated.

The development of the Open-source Software project Open Traffic, specifically aims to provide a modelling toolbox that facilitates researchers to develop customized modelling methods and applications. The basic difference with other open source projects such as the micro simulation model SUMO\(^1\) and the multi-agent Transportation Simulation Toolbox Matsim\(^2\), is the objective to provide a modular meta-modelling environment with utilities that support the development of various modelling approaches. The Open-source approach and the objective to provide model free traffic networks have been described previously\(^3\)\(^-\)\(^5\).

This article describes the high level architecture of Open Traffic and provides some examples to elucidate the specifications of its functional design. The high level structure (or skeleton) of the transport model globally identifies the building blocks with its main objects, the relationships between objects, and the methods and algorithms that describe the functioning of the transport system. Additionally, the possibilities and requirements to adhere Open Traffic to agent based modelling approaches are explored.

2. Considerations for the high level architecture of Open Traffic

Open Traffic provides a modelling framework, that aims to facilitate the modelling of transport demand and the assignment of traffic, as is shown in Fig. 1. For reasons of clarity, feedback loops between these steps are not included in the figure. The subsequent approach of all these modelling steps can vary in many different ways, for example by (1) the type of model application (such as planning, operational or embedded)\(^6\), (2) the element(s) of the traffic system they aim to describe, (3) the scale with respect to time (from minutes to a number of years) and space (segment, lane, link, route, network)\(^7\), and (4) the underlying modelling paradigms used (see\(^8\)\(^-\)\(^9\)). Important considerations for the architecture of Open Traffic are the requirements to enable the joint implementation of multiple modalities (such as car, pedestrian or train) and a seamless integration of variations in scale (level of detail) and size (area), in order to facilitate multi-scale and multi-modal modelling methods. For example, the collection of objects of a road network must be defined in such a way that it enables the creation of networks for a detailed micro simulation, but also for a macroscopic assignment, or the application of network fundamental diagrams (that require areas as an entity). The benefit of this approach is that data objects only need to be defined once, and methods support the creation of networks with different levels of detail.
3. Main components of Open Traffic

The Open Source approach requires an object oriented design, allowing users to focus on a specific component without considering the other components. We therfore create building blocks (in terms of Java named packages) that include a collection of components (classes). These classes specify objects with their attributes and behaviours. To create transport model applications, additional methods specify how these various components are connected. In Open Traffic, the “Model” object acts as the parent entity for all other objects that can be part of the traffic model environment. The main parts of the “Model” are:

- the physical environment; composed of several object groups such as the transportation infrastructure, buildings, terrain and vegetation, water bodies and city and road furniture;
- the population; conducting activities at different locations;
- the movable objects; such as vehicles, mopeds and bicycles;
- activities; carried out by the population;
- traffic demand as the result of activities of people at different locations.

The description of the spatial environment (where movements occur) is one of the basic input elements for traffic models. Currently, the majority of traffic models deploys a specific object design for traffic networks, that is not synchronized with internationally accepted standards. In order to improve this situation we propose a description of the objects that does align to guidelines from the ISO/TC 211, Open Geospatial Consortium (OGC) and CEN/TC 287 are relevant. The definition of Open Traffic objects is based on the philosophy of CityGML, a 3D GIS standard for spatial objects that is accepted by the OGC\textsuperscript{10}. CityGML contains an extensive library with object definitions for...
describing a 3D city and rural environment. City GML allows the definition of objects at multiple levels of detail. At the lowest level of detail (LOD 0), a road can be defined as a link between two nodes. More detail can be added at LOD 1, where a road is described as a shape (polygon). LOD 2-4 provide more details by identifying separate road segments thematically, describing their topology, usage and functionality. At this level we can distinguish segments like a driving lane, parking lots, a barrier or a pedestrian crossing. This level of detail is suitable for defining infrastructure in micro simulations, whereas a lower level of detail can be sufficient for assignments that require less detail.

Fig. 2 shows the collection of packages that describe the transport environment. The GeoObjects package contains the objects that describe the model infrastructure and the physical environment. For road traffic, the network is defined by directed links that are connected through nodes. Additionally, activity locations are the origin and destinations of trips. These locations are connected to the network by means of zones. Zones can be defined as micro-zones or poly-zones, both representing a predefined area within the network area. The micro-zones are defined as points that are connected to the network through one or more zone connectors (links). The poly-zones are (multi)polygons that cover a certain area. The start or end of a trip is on one of the links within this area.

The actual infrastructure is based on the topography of the relevant spatial objects. Fig. 3 shows the main objects that are part of the generic Network object. The link object is specified as a design line directed by a fromNode and a toNode. Vertices allow for a refinement of the link geometry. A link has a list of one or more cross section objects. A cross section is defined by one or more cross section elements, such as the hard or soft shoulder lane, the road surface, barriers and the central reservation area. Multiple cross sections describe variations in the road lay-out. The definition of the road surface is further detailed by road markers. The road marker along describes the division of a road into lanes and provides rules for overtaking, the turn arrows define the allowed turning movements at a junction and stop lines define the position of the entrance of the junction. The lane objects are derived from the lane marking along. Similarly, a turning lane is not defined as an attribute of a road or lane, but instead is derived from a turn arrow object that is defined by its location (x, y). To generate a specific infrastructure that can be used for a certain model application, we provide methods that use these basic objects to derive a coherent network. In this way, multiple types of model networks can be generated.
Traffic demand can be defined very detailed, but allows for aggregation to more global approaches, such as an origin-destination based trip matrix. At the highest level of detail, the \textit{tripPattern} object represents a list of activities, its locations and the preferred starting time of the activity for every \textit{person} object. Lower levels of detail can be attained, for instance by defining \textit{tripPattern} as a list of destination zones, with the preferred starting time per zone and a collection of \textit{person} objects that conduct this trip pattern.

4. Supporting utilities

Open Traffic facilitates the import of both Open Streetmap networks (OSM XML file format) and Omnitrans networks that are exported to the ESRI shape-format. The imported networks are stored in dedicated XML (eXtendable Markup Language) format\textsuperscript{11}. Additionally, ‘utilities’ are provided to manipulate data and objects. An example is the method to derive the geometry of turning movements between entrance and exit lanes of a junction. For networks without a detailed description of the junction geometry, we provide an automated node expansion method. By applying a recursive approach, we create a sub-network for every junction node. Fig. 4 graphically shows the steps of this automated creation of a junction area with lane-connectors for the allowed turning movements. Based on the allowed turns per lane, connecting links and lanes at the junction area are created. Turning restrictions are defined by the absence of lane-connectors at a junction.

The Open Traffic graphical editor can be used to visualise and edit objects from a map. We have ‘outsourced’ parts of the developments by implementing Open Source Software library\textsuperscript{12}: the Java Topology Suite (JTS) and GeoTools, both Open Source Libraries for Java. JTS conforms to the Simple Features Specification for SQL\textsuperscript{13},
which implies that the basic geometrical entities are the same in the Java environment as in GML (Geographical Marking Language) and that there is no need for geometrical transformations during the import of the data into the Java environment. Secondly, it contains robust implementations of the most fundamental spatial algorithms (in 2D)\textsuperscript{14}. Additionally, the GeoTools toolbox is included for its mapping capabilities, data parsing for a broad range of spatial data formats and spatial database access\textsuperscript{15}. The graphical editor visualizes the Open Traffic transport infrastructure, and can easily be extended with methods to show application specific networks. The networks can be edited graphically by moving, adding, inserting and deleting objects. The attribute data of any object (such as the speed or street name) can be inspected and edited by selecting an object and changing the data in an edit box (see Fig. 5). This is implemented using the Java programming language feature called \textit{reflection}\textsuperscript{16}. This enables the creation of class browsers, debuggers and test tools entirely within the language in a way that works on all computers that run Java programs. In addition to the graphical editor, the Open Source library JFreeChart is included for the presentation of data in charts and plots for instance for presenting trajectory data.

5. Methods and algorithms

Open Traffic contains methods and algorithms that can be used as building blocks for a traffic model. This part of Open Traffic is meant to be the essence for target users. Currently, the main tools and methods concern the import and conversion of geographic objects into functional road traffic networks. Whereas the \textit{utilities} are meant to support the user, the methods and algorithms section is the part where developments and innovations are intended to take place. By combining methods and algorithms in a workflow, model applications can be generated for demand generation and/or assignment. Newly developed modules can be stored in Open Traffic and evolve into a library of advanced modelling methods. The architecture of this library approach will be developed simultaneously with the implementation of new functional modules. A first example is the implementation of the Microscopic Open Traffic Simulation (MOTUS), a simulator that explicitly provides opportunities to simulate ITS applications. The Java application MOTUS is developed as a stand-alone adaptable and extendable research framework and is implemented as a \textit{simulator} in OpenTraffic\textsuperscript{17}. By structuring the various driving tasks in the simulation, different models for the same task can be interchanged and new models can easily be developed\textsuperscript{18}. This aligns to the modular approach of Open Traffic.
The two main sources of input data are the traffic network and traffic demand. The Open Traffic network design appears to be suitable for MOTUS. A relatively simple set of conversion methods enables the creation of a network that satisfies the MOTUS requirements. The objects that are created are exported to an intermediate data file, which enables the inspection and validation of the converted network. The export comprises the links and lanes, their attributes such as speed limit, length, lane marking definitions and conflicts at merges and junctions. Traffic demand in MOTUS is defined as the number of trips by O-D relation and linked to lanes by their variables origin and destination. This directly connects to the Open Traffic definitions of traffic demand in the TripPattern object and requires no additional intelligence. Routes can both be generated within MOTUS, or connected to the Open Traffic path finding algorithms. This connection is implemented by transferring the paths to MOTUS formats. Again, a simple procedure allows this transfer in a very straightforward manner. The simulation control in Open Traffic is implemented in a basic version and currently allows for running the micro simulation with run and stop commands. The graphical editor visualizes the movement of vehicles per time step. Every vehicle can be selected by the ObjectInspector which provides information of the vehicle/driver combination, such as the leading vehicle, position at the lane, speed and acceleration, as can be seen in Fig. 5.

6. Enabling agent based approaches

The agent and activity based approach emphasizes the modelling of human behaviour: an ingredient that is still lacking in most transport models. While the architecture of Open Traffic takes account of the requirements for activity based models, for instance by connecting activity plans to traffic demand, the current version of Open Traffic mainly focusses on the assignment and simulation part of transport models. To truly enable agent based approaches within Open Traffic, further developments of objects and methods are required, specifically addressing the behaviour of agents in relation with its environment, enabling the coupling with agent based simulation models such as Janus or JaSim\textsuperscript{[4]}\textsuperscript{[5]}\textsuperscript{[20,21]. The holonic metamodel for agent oriented analysis and design, profoundly identifies the methodology to incorporate the organizational and holonic concepts into multiagent software systems\textsuperscript{[20,21]. The approach explicitly models the role that agents play within their specific context, regarding both the physical and the communication environment. An example from the field of transport would be the difference in driving behaviour during a normal day (relaxed) and in exceptional circumstances, for instance an evacuation (stressed). With the upcoming vehicle technologies leading to an increase of interactions and communication between vehicle/driver entities (and its environment), the multiagent based modelling concepts seem worthwhile to be further adapted in traffic simulations.

7. Conclusion and Outlook

The development of the Open Traffic toolbox for traffic modelling methods is based on a flexible and modular approach, which requires a high quality of both the software architecture and the design of the objects and methods. As a first test case the micro simulation model MOTUS has been successfully implemented in Open Traffic. The development of the necessary conversion modules appeared to be relatively straightforward. A mature first release that can be downloaded and used by external users is planned in the Summer of 2015. This will be a joint development of both MOTUS and Open Traffic that results in a simulation lab with tools to create, edit and analyse traffic models at different simulation scales (macro, micro), different transportation modes and various methods and algorithms in a user-friendly software environment. Currently, Open Traffic is only one out of a group of open source developments. A closer collaboration with other open source initiatives would be a useful step to synchronize and stimulate further modelling developments.

Acknowledgements

The work presented in this article is partly supported by the Dutch Foundation of Scientific Research MaGW-now under the research program “Traffic and Travel Behavior in case of Exceptional Events”.
References

1. Krajzewicz, D., M. Bonert, and P. Wagner, The Open Source Traffic Simulation Package SUMO, 2006.
2. Balmer, M., et al., MATSim-T: Architecture and Simulation Times. 2009.
3. Tamminga, G., et al. Design of an Open Source Traffic and Travel Simulation Framework. 2012.
4. Miska, M., et al. OpenTraffic - An Open Source Platform for Traffic Simulation. in Australasian Transport Research Forum 2011.
5. Tamminga, G., et al. Toward GIS-Compliant Data Structures for Traffic and Transportation Models. in Transportation Research Board 92nd Annual Meeting. 2013.
6. van Noort, M., B. van Arem, and P. Byungkyu. MOBYSIM: An integrated traffic simulation platform. in Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on. 2010.
7. Barceló, J., Models, Traffic Models, Simulation, and Traffic Simulation, in Fundamentals of Traffic Simulation, F.S. Hillier, Editor 2010, Springer New York. p. 1-62.
8. Hoogendoorn, S.P., State-of-the-art of vehicular traffic flow modelling. Journal of Systems and Control Engineering, 2001. Vol 215 No 14.
9. Arentze, T. and H. Timmermans, Albatross: a learning based transportation oriented simulation system2000: Eirass Eindhoven.
10. Consortium, O.G., OpenGIS® City Geography Markup Language (CityGML) Encoding Standard T.H.K. Gerhard Gröger, Claus Nagel, Karl-Heinz Häfele, Editor 2012.
11. Salminsh, K. XML applications in SLR. in Proceedings of the 13 th International Workshop on Laser Ranging. Washington, USA. 2003.
12. Haelfiger, S., G. von Krogh, and S. Spaeth, Code Reuse in Open Source Software. Management science, 2008. 54(1): p. 180-193.
13. Consortium, O.G., OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 2: SQL option, 2010.
14. Johansson, M. and L. Harrie. Using Java Topology Suite for real-time data generalisation and integration. in Proceedings of the 2002 workshop of the International Society for Photogrammetry and Remote Sensing, 2002.
15. Turton, I., Geo tools. Open Source Approaches in Spatial Data Handling, 2008: p. 153-169.
16. [http://docs.oracle.com/javase/tutorial/reflect/index.html].
17. Yuan, Y., A.J. Pel, and S.P. Hoogendoorn, From normal driving behaviour to emergency driving behaviour: the impact of the transition phase on traffic flow operations during evacuation. . Submitted to EURO Journal on Transportation and Logistics (under review).
18. Schakel, W., et al. A Modular Approach for Exchangeable Driving Task Models in a Microscopic Simulation Framework. in IEEE-ITSC. 2013. The Hague.
19. Galland, S., et al. Environment model for multiagent-based simulation of 3D urban systems. in the 7th European Workshop on Multi-Agent Systems (EUMAS09), Ayia Napa, Cyprus. 2009.
20. Cossentino, M., et al., A holonic metamodel for agent-oriented analysis and design, in Holonic and Multi-Agent Systems for Manufacturing2007, Springer. p. 237-246.
21. Galland, S., et al., Multi-Agent Simulation of Individual Mobility Behavior in Carpooling using the Janus and JaSim Platforms. Transportation Research Part C, 2013.