TraffSim -- A Traffic Simulator for Investigating Benefits Ensuing from Intelligent Traffic Management

Conference Paper · November 2013

DOI: 10.1109/EMS.2013.76

3 authors:

Christian Backfrieder
Fachhochschule Oberösterreich

Christoph F Mecklenbräuker
TU Wien

Gerald Ostermayer
Fachhochschule Oberösterreich

2013 European Modelling Symposium

TraffSim - A Traffic Simulator for Investigating Benefits Ensuing from Intelligent Traffic Management

Christian Backfrieder*, Christoph F. Mecklenbräuker† and Gerald Ostermayer*
*RG Mobile Radio Networks, UAS Upper Austria, Hagenberg, Austria
Email: christian.backfrieder@fh-hagenberg.at, gerald.ostermayer@fh-hagenberg.at
†Christian Doppler Lab Wireless Technology for Sustainable Mobility
Vienna University of Technology, Vienna, Austria
Email: cfm@nt.tuwien.ac.at

***Abstract*—With our interest to reduce traffic congestion, simulations are necessary to investigate how to achieve this goal. Because of the difficulty of real-time experiments, traffic simulators are a good and widely used tool for analyzing specific situations of the traffic flow virtually. This paper introduces TraffSim, a new platform-independent framework, which is capable of simulating microscopic vehicular road traffic in a time-continuous manner. It is able to simulate scenarios close to reality by using OpenStreetMap networks, creating precise and at the same time reproducible results. TraffSim is able to use various network types from multi-lane motorways to dense city road networks including multiple types of junctions or one-way streets and provides great extensibility and flexibility for both developers and users. Moreover, it allows free definition of vehicle properties like dimensions or engine features, which are used by several applicable models for longitudinal movement, lane-changing and fuel-consumption. We present the architectural and logical aspects of the simulator, as well as the user interface and statistics features.**

***Keywords*-road traffic, traffic simulation, jam, congestion, traffic management, time-continuous simulation, vehicle routing, junction, open street map**

I. Introduction

Modeling of vehicular road traffic is a very complex topic. It has serious consequences for human drivers, residents of roads, animals and the environment how we use vehicles to travel from A to B. The volume of traffic increases steadily, especially in bigger cities and areas of high population density. Therefore, it is essential to deal with the big amount of vehicles optimally to minimize the driving time, travelled distance and subsequently exhausted emissions. Traffic jams cause, in addition to the mentioned issues, huge public cost for cities or countries, and should therefore be reduced. For that reasons it is important to learn and understand how to deal with this problem, and to achieve an improvement by intelligent control and management of traffic.

In order to investigate the influence of traffic management, simulators are a practical instrument for trying to predict the impacts of different deployed applications. Different situations and parameters can be put into them with the objective of prognosticate possible implications resulting of the input change. Simulators can mainly be classified in microscopic or macroscopic types, depending on the granularity of investigation and temporal resolution. Macroscopic models deal with the traffic state as a whole with attributes like traffic flow, density or average speed, while microscopic models analyze the single vehicles as single particles in the system. Each simulated unit has its counterpart in reality. The TraffSim simulation framework is a microscopic simulator, calculating speed, acceleration, fuel consumption, travel distance or time among other parameters for each modeled vehicle separately.

TraffSim was developed with the idea of an extensible, freely available simulation framework, which is fully configurable in advance by XML input files. In order to investigate and learn from the impacts of different parameter changes, it is very important to allow reproducibility in simulations. TraffSim is able to re-run a simulation as desired and creates equal output when using the same input configuration. The input configuration includes definition of the road network, junction control, vehicle types, car-following models for longitudinal movement, lane-change models and routes to define how vehicles intend to travel to their target. Furthermore, fuel-consumption is calculated to investigate the emissions exhausted by the vehicles.

The rest of the paper is organized as follows. In the next section, comparable projects from literature are elaborated and requirements for TraffSim are defined. The third section deals with the basic framework itself, and introduces the Java architecture of the simulator as well as implemented models. Furthermore, the Eclipse RCP user interface is presented in detail. In the fourth section, applications of the simulator are explained. Section V concludes the paper and gives and overview of planned future development.

II. Related Work

Several projects dealing with traffic simulation exist in literature, which all have some strengths in a specific area and also differ in other aspects. The following traffic simulation projects were picked out, and we want to give a short overview about the features and qualities of them.

978-1-4799-2578-0/13 $31.00 © 2013 IEEE
DOI 10.1109/EMS.2013.76

A. Existing Projects

A very powerful simulator is *MovSim* in [1], which was initiated at the TU Dresden as platform independent Java open-source project. It is a microscopic simulator, which has its strengths in having implemented several different models for car-following, using its own fuel-consumption model and lane change model MOBIL (discussed in [2]). Nevertheless, it has no importer for OpenStreetMap (OSM) networks, and also no support for junctions (meaning not motorway ramps, but junctions with multiple in- and output lanes).

Another huge microscopic and multi-modal traffic flow simulation framework is *VISSIM*, which is developed by PTV Planung Transport Verkehr AG in Karlsruhe, Germany. It is used worldwide for microscopic traffic simulations in cities and able to simulate different vehicle types, but also public transport, pedestrians and bicyclists. VISSIM is running under windows and licensed commercially.

The *Traffic Simulation Framework* explained in [3] is a cellular automaton (CA)-based simulation model, which further is an extension of the Nagel-Schreckenberg model [4]. It is implemented for Windows platforms using the C# programming language. CA simulators split the physical location into cells of length Δx, which can hold the state occupied or free for each timestamp. In addition, more attributes such as speed can be assigned to occupied cells. They are often faster than continuous simulators, but due to the fact that no fuel consumption model is implemented in this framework and CA models often lead to unrealistic values or unpredictable outcomes, which are among other things a result of stochastic terms [5], it does not fit the needs defined for TraffSim.

A further microscopic and macroscopic free-flow traffic simulator is *FreeSim* [6]. It is platform independent and describes itself as being easily extensible. FreeSim models vehicles basically as self-managed entities, which can send and receive information via a central server and traverse along freeway segments. It allows monitoring of travel times and congestion through the centralized server, but is mainly designed for wide-ranging motorway networks rather than dense cities.

B. Requirements for TraffSim

TraffSim was implemented for investigating a defined topic within the research project SmartTraffic, executed at the UAS Upper Austria. Its main scope is to find out how intelligent management of traffic flows can reduce congestion and traffic jams. This is done by assuming a working communication between the cars and infrastructure, which is then used to smartly re-route cars to bypass possible traffic jams. The gained benefits from the fast and early information propagation, compared to radio broadcast as well as the performance of the re-routing mechanism, can be measured for example in less fuel consumption, less time consumption or lower CO_2 emissions.

For achieving this goal, the following requirements were defined for TraffSim:

- fully reproducible results
- realistic longitudinal models
- fuel consumption model
- usage of real road networks (OSM)
- cleanly encapsulated architecture
  - full abstraction of the features
  - well defined interfaces
  - possibility for extensions without any kernel code changes
- interactive control for developer's self-learning during implementation and fact-finding phase
- junctions must be modeled
- crash detection
- time and date attributes for modeling peak hours or other time-dependent situations
- time and location-continuous simulation for precise results
- platform independent
- parallel simulations of multiple scenarios
- performance optimized through multi-threading

All the aggregated functions provided by TraffSim are necessary to perform the investigations planned within the SmartTraffic research project. None of the described simulators includes those required functions without limitations. TraffSim combines the features and strengths of existing simulators which are required for fulfilling these defined requirements. It further extends functionality by routing possibilities or implementations of junction logic.

III. The TraffSim Framework

A. Environment

This application is implemented as Eclipse Rich Client Platform (RCP) application, which brings several benefits for both developers and users. First of all, it is able to deploy to a variety of desktop operating systems, such as Windows, Linux and Mac OSX. It provides an automated update mechanism, is built up through a plugin framework, which easily enables adding or exchanging components and uses the commonly known Eclipse UI, including the view concept, preference screen or toolbars [7]. This is certainly a benefit for users which are already familiar with other Eclipse products as well.

B. Architecture and Components

The modules of TraffSim can be divided as shown in the component diagram in figure 1.

The Kernel is the central manager of the application, which creates multiple simulation models. These hold all necessary information about a simulation and read the configuration from the input files. The crash detector, statistics

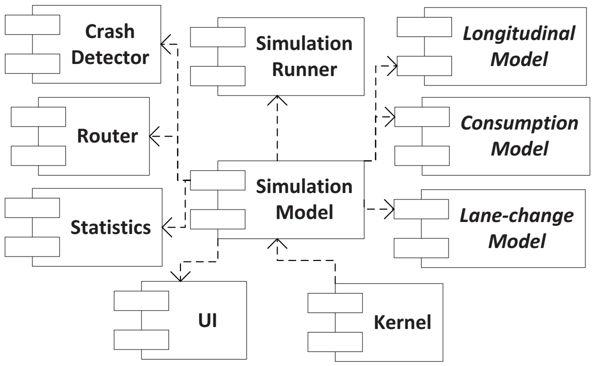


Figure 1. TraffSim architecture

component, router and all required models are set up by this component as defined in the configuration. TraffSim is a time- and location continuous simulator, which means that vehicle positions are available for any arbitrary timestamp. The simulation runner is responsible for driving the simulation forward by a configurable interval. This mechanism is implemented event-based, meaning that all components interested in time updates have to be registered within the simulation runner, and get subsequently notified about updates. They do their job, and wait for the next update. The user interface (UI) component also shows separate windows for each created simulation, besides displaying an environment container for the simulations where users can interact and for example load existing configurations, show, hide or move views, display details about any entities or pause a running simulation.

So as to optimize performance, concurrent execution in multiple Java threads is used. This asynchronous execution has also disadvantages in requiring more effort for synchronizing threads and keeping commonly used data consistent. Moreover, the results of a simulation must not depend in any form on the operating system's scheduler. In TraffSim, the UI is implemented asynchronously in order not to block the running simulation while updating. It paints the current state as often as possible. Some statistic graphs can be updated continuously, which is also executed in parallel to the main worker. The worst case when having too less CPU power is a lower UI frame rate, which can be avoided by increasing sleep time between simulation time updates. The crash detector can be configured to run asynchronous as well, however, this is not recommended in simulations which should produce scientific results because there is a low probability for overlooking crashes, but only during testing and developing phase.

In consequence of the abstraction and separation of the components it is also possible to run multiple simulations in parallel and gain knowledge about impacts of parameter changes visually. Each simulation is executed in a sandbox and has its own environment without any mutual manipulation, as illustrated in figure 2.

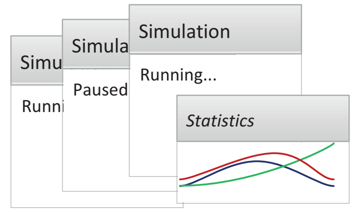


Figure 2. Parallel simulations

C. Data Structure

Moving entities in TraffSim are called vehicles, which can be set up individually to represent small cars, estate cars or trucks by freely configurable lengths, widths, various models or engine.

The static infrastructure is reproduced virtually through road segments, which is a part of a street in a single direction and owns its geometry, a speed limit and an id for routing. It supplementary contains lane segments, which correspond to the number of lanes in one direction. The third main data type is a junction, which connects two or more road segments. It also contains lane connectors, which connect each lane inwards to each possible lane outwards and define the vehicle's tracks through the junction, as depicted in figure 4.

D. Input Structure

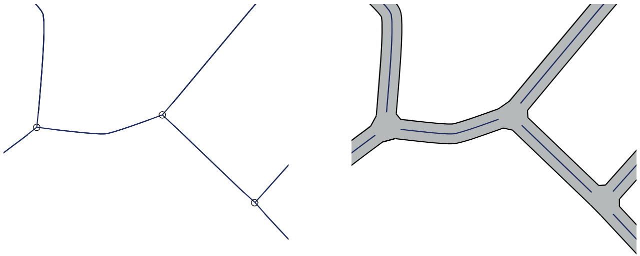
Each simulation uses multiple XML files for defining all necessary input parameters. These are referenced in a main configuration file, which is can be loaded within the UI by the user. An example for a definition of a vehicle is available in listing 1.

```
<Vehicle id="1">
  <label></label>
  <vehicleType>Car</vehicleType>
  <length>4.77</length>
  <width>1.82</width>
  <laneChaneModel>MOBIL1</laneChaneModel>
  <longitudinalModel>ACC1</longitudinalModel>
  <consumptionModel>Diesel_90kw</consumptionModel>
  <routeId>127</routeId>
  <startDate>2013-10-10 15:21:19.216 UTC</startDate>
  <initialSpeed>20.0</initialSpeed>
</Vehicle>
```

Listing 1. Vehicle configuration

E. Road Network

1) Road Segments: The road network is stored in a similar XML format to vehicles. It contains OSM geometry data, augmented with lane geometry and right and left edges of the roads for nice visual presentation. TraffSim allows smoothing of very sharp curves and creation of a specified number of lanes by shifting the original geometry by the lane width and support for entry and exit lanes. The mentioned road segments are connectors between OSM nodes, which



(a) OSM (b) TraffSim

Figure 3. Network snippet as original OSM segments (3a) and TraffSim-postprocessed roads with generated junctions and lanes (3b)

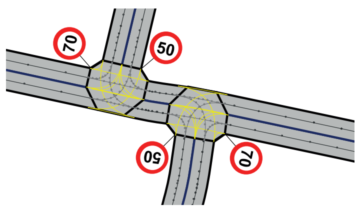


Figure 4. Detail view of junctions with visible speed limit signs, the connecting lanes inside and road lanes outside

are road crossings. Of course, junctions are areas rather than simple points where lines are connected. Therefore, road segments are shortened by the junction radius and assigned connected to the junction. OSM data often contains very short road segments, which are removed if they disappear during the shortening procedure, and of course all routes referencing them are also updated. Figure 3 shows the network before and after TraffSim's postprocessing.

2) Junctions: Road crossings are modeled as separate entities, which handle all vehicles approaching, waiting, inside and outside, but already granted to enter the junction. The logic is encapsulated by an abstract base class, which specifies the interface for implementing objects. They provide an interface method to determine whether or not a specific vehicle v can enter the junction from lane $L1$ inwards to lane $L2$ outwards. The junction implementation itself has to lead the vehicles through in a way close to reality and avoid crashes.

Implementations for unregulated junctions without any right of way and stop signs applying European rules (right before left) and junctions regulated with traffic lights are already available within TraffSim and can also be configured by the input configuration.

F. Models

1) Longitudinal Models: Those models deal with the vehicles propagation along the road. Within TraffSim, they are implemented as time continuous car-following models, according to [8]. To be more precise, the current simulator includes an intelligent driver model (IDM) and a model for adaptive cruise control (ACC), which is similar to automated driving enriched with some human aspects.

Those models must be able to calculate the acceleration for a given vehicle on a given lane of the road, including important attributes as current speed, current acceleration, speed of front vehicle and speed limit.

2) Fuel-Consumption Models: Basically, fuel consumption is a very good argument when talking about saving costs as a result of intelligent traffic management, not only because it also implies a reduction of emissions. Therefore, the simulation framework must provide methods to calculate the instantaneous fuel consumption, as well as summing it up and relate it with travelled time and kilometers.

TraffSim provides an implementation of the microscopic model, which uses the current speed and acceleration values as dynamic input. Aside from that, it uses engine attributes as the cylinder volume, power of the engine, idle and maximum rotation rates and gear ratios as well as physical car data like the car mass, cross section surface or the dynamic tyre radius for the calculation. A very detailed explanation of this model can be found in [10] and references therein.

3) Lane-Change Models: For roads with multiple lanes a model for the decision of lane changes is essential in a simulator. It is anything but trivial to develop a model which is crash-free and considerates other vehicles politely and realistic at the same time. During research, we had a deeper look in the model used in [1], and it fits the requirements of TraffSim sufficiently. The general model is called "Minimizing Overall Braking Induced by Lane Change (MOBIL)" and elucidated in [2]. This model does not only take the risk for the lane-changing vehicle into account, but also considers effects of neighbor vehicles such as the old and new follower or old and new leader in terms of longitudinal accelerations, calculated by the assigned car-following model. Thus, it is also possible to perform lane changes for allowing an entering vehicle to leave the ramp and merge into the continuing lane.

G. Statistics

A very important functionality is the recording of interesting data for analysis after finished simulations. TraffSim records data for vehicles, roads and junctions automatically for live display or postprocessing, which includes speed, acceleration, overall travelled distance and fuel consumption for vehicles and the vehicle throughput including time, vehicle id, type and speed for each road segment and junction.

H. Routing

Vehicle routing is a very important topic for TraffSim, since scenarios are intended to be reproduced with different parameter sets, but the same initial routes for the defined vehicles. To be exact, each vehicle has its predefined source, target and route from source to target. For the junction

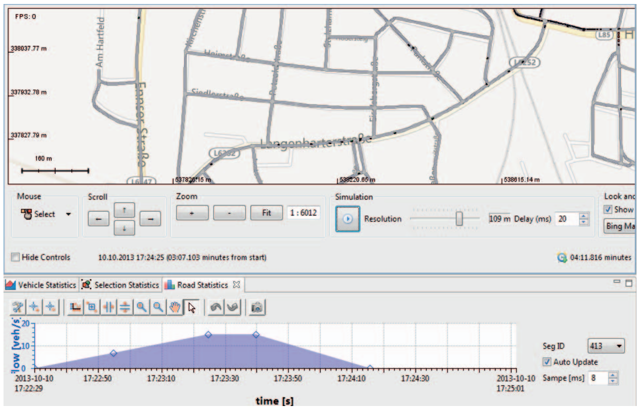


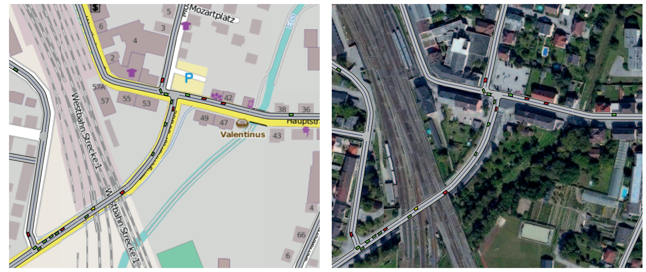
Figure 5. Main overview of TraffSim's UI

controls described in III-E2 to work properly, vehicles must know where they would like to go when asking a junction for permission to enter. The junction accordingly leads the vehicle to the desired connector to continue its route. Generally, a route is a list of ids which are to be passed by a vehicle to reach its destination. Each road segment owns a routing id. The routes are generated prior to simulation by external tools and saved within the TraffSim configuration. The framework already provides an implementation for route creation, which is very simple but works like a charm. It finds loose edges in the network randomly, chooses a source and target road segment and creates a desired amount of routes. The project *Osm2po* in [9], which is able to create routes, is used for this purpose. It uses a weighted graph, created out of OSM XML data and contains multiple routing algorithms based on Dijkstra/AStar. Osm2po can be included very easily by just one Java library, and is therefore also platform independent and does not need any extra installation.

I. User Interface

The UI is built up modularly, using eclipse RCP views for the different displayed content, as figure 5 shows. The main area includes an overview of the current state of the simulation with the road network and all vehicles inside it. It allows zooming and translation to all directions, as known from digital map applications like OSM. Moreover, maps from various providers can be used as background layer, including OSM (fig. 6a), Google Maps or Bing Maps for road display or also satellite or hybrid (6b) layers. This allows users to gain a better orientation for known locations, and also enables validation of the loaded road network.

The lower part of the main window is used for statistic views by default. But the RCP environment allows moving, relocating or even detaching of views as desired, so that users can build up their layout as they like to. All other views (setting up traffic lights, exporting the map, finding items, details view, ...) are opened detached by default, since



(a) OSM (b) TraffSim

Figure 6. Road network display with OSM layer or hybrid satellite view

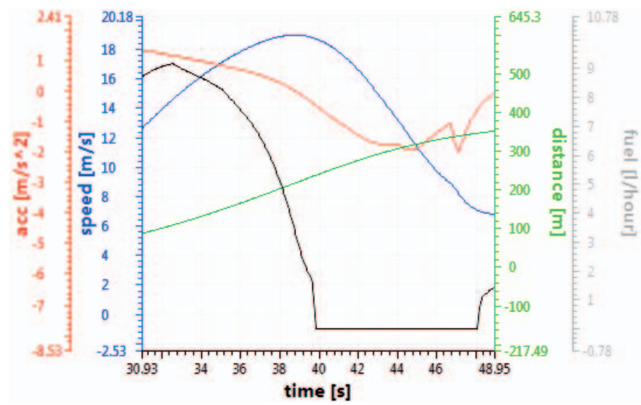


Figure 7. Vehicle statistics view

they are usually not used permanently.

TraffSim allows live inspection of entities within the simulator, such as vehicles or junctions for example. Vehicles are monitored by a global vehicle statistics view, which allows selection of the vehicle id and displays curves from the past (fig. 7). Junctions and their logic of managing the vehicle flow can be monitored in junction detail windows. Latter can be opened and inspected simultaneously for multiple junctions by detaching from the main window and moving them around the screen.

IV. Simulation Applications

TraffSim can be used for various situations where microscopic traffic simulation is necessary and useful. Any research based on vehicular traffic using travel times, fuel consumption or average speed is applicable and relatively easy to configure within TraffSim. Simulations for multi-lane highways including entrance and exit ramps have been already executed, as well as for dense road networks within cities. Within this research project, TraffSim will be used to simulate real environments and find out how travel time, fuel consumption and travel distance change if vehicles get information very early and bypass congested road sections.

V. Future Work

TraffSim is still under development. The following enhancements and extension have been thought of or are planned in the near future:

A. Live Re-Routing of Vehicles

With the purpose of changing a vehicles route so that delays can be avoided as good as possible, calculation of a new route which is an alternative to the current one whilst bypassing the congested road segment(s) is necessary. This could be implemented by using the *Osm2po* library, which is already in use for pre-calculation of the routes. First, the routing graph must be updated temporarily in a way that congested road segments refer to a cost value corresponding to the calculated real-time delay. Second, a new route must be associated to the affected vehicle(s). We already did a first case study on this topic, which achieved satisfying results and showed very good routing performance.

B. Intelligent Detection of Traffic Congestion and Jams

To make use of the re-routing feature, it is favorable to detect congested road segments automatically and react by bypassing these segments.

C. Consideration of Terrain Geometry

Hills and slopes have considerable impact on fuel consumption, as indicated in [11]. For a more precise calculation of fuel consumption, it is therefore suggestive to use altitudes in the imported map and include them in the calculation. This data could for example be mapped from Shuttle Radar Topography Mission (SRTM) from [12], which is available for free in a resolution of about 90 meters almost all over the world.

D. Intelligent Route Creation

The source and target of vehicles should be found via intelligent heuristics rather than randomly. For example, the population density or traffic measurements can be used to identify the routes.

VI. Conclusion

Simulators are very important components of intelligent traffic management systems. Design and implementation of them is a challenging problem, which is solved by this paper. We present TraffSim, a microscopic, parallel and time-continuous traffic model for simulating real traffic in real environments. It allows easy comparison of input parameter changes by parallel simulations and generates fully reproducible results, whilst also making use of multithreading for achieving good simulation performance. Its architecture is designed modularly to encapsulate the functionality and provide mechanisms for extension and adaption of features. Implementations of car-following models for longitudinal movement, a fuel-consumption model, a lane-change model as well as junction controls for unregulated and traffic-light controlled junctions are built in. A concurrent crash detection ensures safe and realistic behaviour. The statistics component records appropriate data for vehicles, road segments and junctions for result analysis in postprocessing. The user can control and observer simulations through a representative Eclipse RCP user interface.

Acknowledgment

This project was supported by the program *Regionale Wettbewerbsfähigkeit OÖ 2010-2013*, which is financed by the European Regional Development Fund and the Government of Upper Austria.

References

[1] M. Treiber and A. Kesting, "An Open-Source Microscopic Traffic Simulator", IEEE Intelligent Transportation Systems Magazine, vol. 2, no. 3, pp. 613, 2010

[2] A. Kesting, M. Treiber and D. Helbing, "General lane-changing model MOBIL for car-following models", Transportation Research Record: Journal of the Transportation Research Board 1999.1, pp. 86-94, 2007

[3] P. Gora, "Traffic Simulation Framework", 14th International Conference on Computer Modelling and Simulation (UKSim), 2012, pp. 345349.

[4] K. Nagel and M. Schreckenberg, "A cellular automaton model for freeway traffic", Journal de Physique 2, 1992, pp. 2221-2229

[5] M. Treiber and A. Kesting, "Cellular Automata" in Traffic Flow Dynamics: Data, Models and Simulation, Springer, 2012

[6] J. Miller and E. Horowitz, "FreeSim - a free real-time freeway traffic simulator", IEEE Intelligent Transportation Systems Conference, ITSC 2007, 2007, pp. 1823

[7] The Eclipse Foundation. (2013, Oct. 10). Rich Client Platform [Online]. Available: http://wiki.eclipse.org/index.php/Rich_Client_Platform

[8] P.G. Gipps, "A behavioural car-following model for computer simulation", Transportation Research Part B: Methodological, Elsevier, vol. 15(2), pages 105-111, 1991

[9] C. Moeller. (2013, Oct. 14). Routing On OpenStreetMap [Online]. Available: http://www.osm2po.de

[10] M. Treiber and A. Kesting, "Fuel Consumption Models" in Traffic Flow Dynamics: Data, Models and Simulation, Springer, 2012

[11] E. Hellström, M. Ivarsson, J. Åslund, and L. Nielsen, "Look-ahead control for heavy trucks to minimize trip time and fuel consumption", Control Engineering Practice, vol. 17, no. 2, pp. 245254, Feb. 2009

[12] The CGIAR Consortium for Spatial Information (2013, Oct. 15). SRTM 90m Digital Elevation Data [Online]. Available: http://srtm.csi.cgiar.org/