Archives of
Transport System
Telematics

Volume 3

Issue 2

May 2010

GIS-based traffic flow simulation visualization for TRANSIMS

M. MACIEJEWSKI
Institute of Machines and Motor Vehicles, Poznan University of Technology, ul. Piotrowo 3, 60-965 Poznan, Poland
EMAIL: michal.maciejewski@put.poznan.pl

ABSTRACT

The paper presents two visualization applications that enable to visualize traffic flow simulation results obtained with the TRANSIMS system. The applications, designed and implemented at the Institute of Machines and Motor Vehicles of Poznan University of Technology, are written in Java and, as freeware, can be used free of charge. Both visualizers use GIS-based data manipulation techniques to provide precise and customizable 2D visualization.
First, the problem of traffic flow modeling and simulation, and the TRANSIMS system were introduced. Then a general design and features of both applications were described. In order to demonstrate the most essential capabilities of the tools, the author presented an example of visualization of traffic in a segment of a real urban network.

KEYWORDS: traffic flow, urban, model, simulation, microscopic, visualization, TRANSIMS, GIS, Java, GeoTools, FFmpeg

1. Introduction

Visualization of microscopic traffic flow simulation is indispensable in order to evaluate, analyze and verify simulation results. Without detailed visualization it is almost impossible neither to look into the details of traffic flow dynamics nor to observe and assess vehicle-vehicle or vehicle-environment interactions. Unfortunately, not every microsimulator offers visualization tools; it is also the case of TRANSIMS. The aim of the author was to implement tools that will provide highly detailed visualization for TRANSIMS.

2. TRANSIMS

TRANSIMS (TRansportation ANalysis and SIMulation System) [1] is a free integrated simulation system that enables a regional analysis of transportation systems. It supports the whole process of transportation modeling and simulation, from population synthesis through activity generation to traffic microsimulation. The process is usually run iteratively in order to obtain system's equilibrium, according to Wardrop's first principle [2]. Additionally, it is possible to estimate emissions on the basis of microsimulation results.

TRANSIMS consists of several modules, each one responsible for a certain stage within the modeling and simulation process flow (Fig.1):

- Population Synthesizer – creates a population of synthetic individuals according to detailed demographic statistics and other data sources.
- Activity Generator – generates daily activity plans for each synthetic member of the population; daily plans consisting of work, shopping, school, and other kinds of activities.
- Route Planner – chooses transport modes and plans routes for individuals on the basis of previously generated daily plans.

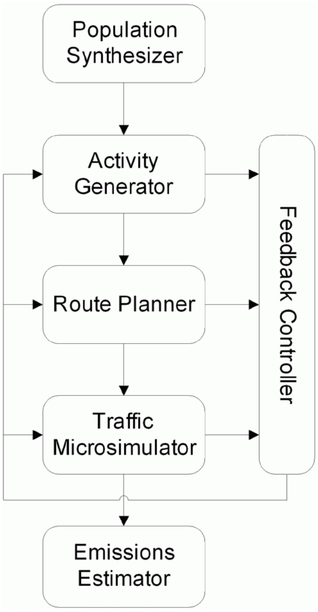

Fig. 1. Modeling and simulation process flow in TRANSIMS
Source: [own work]

- Traffic Microsimulator – simulates the traffic according to the planned routes of the synthetic individuals.
- Feedback Controller – manages the dataflow within the framework; controls the relaxation process by running the former three modules iteratively in order to find the system equilibrium.
- Emissions Estimator – estimates the emission of selected exhaust pollutions.

This research was carried out with TRANSIMS version 4.0.6.01 (containing Traffic Microsimulator module version 4.0.75).

2. Current visualization approaches for TRANSIMS

TRANSIMS is a complex modeling and simulation system with many features; however, it lacks a graphical user interface that would enable users to build network models, or at least to visualize simulations. Although the early versions of TRANSIMS were provided with a simple visualization tool, the current version (4.x) does not have such a tool.

As visual verification and validation of simulation models is essential, Xuesong Zhou has created NEXTA (Network Explorer for Traffic Analysis), "a graphical user interface to facilitate preparation, post-processing and analysis of simulation-based dynamic traffic assignment datasets" [3]. With this software one can run visualization of traffic flow, as shown in Fig. 2. However, due to the one-minute time step, this visualization is of limited functionality, because it does not enable to look into the details of traffic flow dynamics or to observe and assess vehicle-vehicle or vehicle-environment interactions.

3. GIS-based visualization for TRANSIMS

3.1. GIS data

As mentioned before, TRANSIMS does not provide tools for direct visualization of the results; however, it offers a set of simple programs for generation of shapefiles (ESRI Shapefiles). These shapefiles can then be used as layers in various GIS systems. From among them, from the perspective of the simulation visualization, the most interesting programs are:

- ArcNet – creates shapefiles containing different kinds of network data (i.e. nodes, links, shapes, lane-use, activity locations, parking, pocket lanes, lane connectivity, etc.); each shapefile concerns one specific type of data.
- ArcSnapshot – creates shapefiles from selected records in a TRANSIMS snapshot file; each shapefile concerns a distinct time during the simulation. A snapshot file is a kind of a register that contains a detailed history, second by second, of each running vehicle.

The process of creating both types of shapefiles is shown in Fig.3.

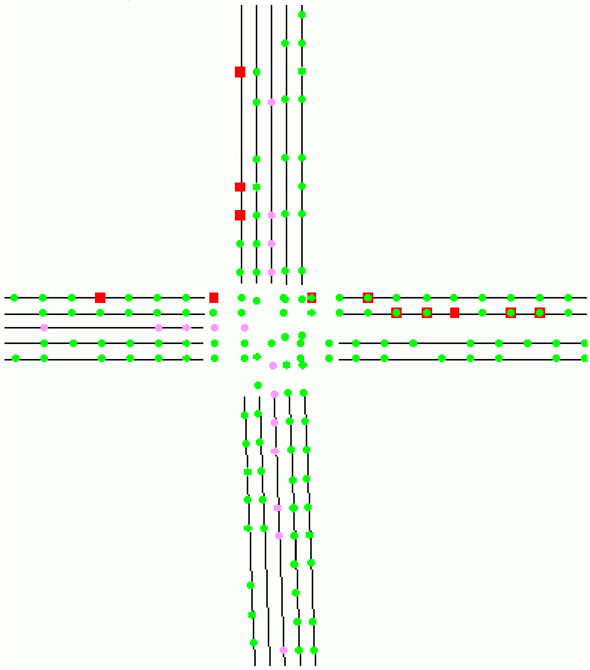

Fig. 2. NEXTA for TRANSIMS
Source: [own work]

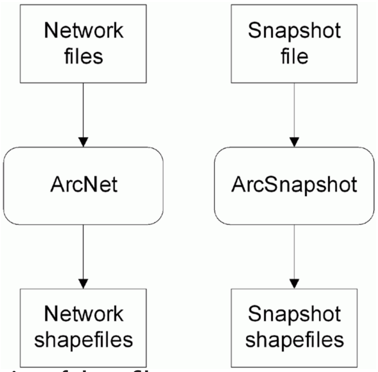

Fig. 3. Creation of shapefiles
Source: [own work]

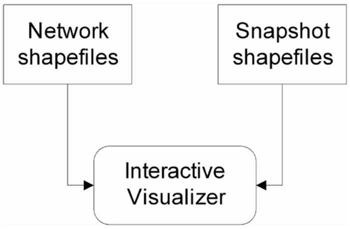

Fig. 4. Interactive visualization
Source: [own work]

3.2. Interactive visualization

By combining selected network shapefiles with a snapshot shapefile, one can visualize a simulation state for any arbitrary chosen simulation second. This can be done by means of almost any GIS system that supports reading of ESRI Shapefiles. Unfortunately, this will only constitute a static picture for a specific moment of the simulation, which is far from the real visualization.

In order to provide an interactive dynamic visualization, the author developed an application called TRANSIMS Visualizer. As shown in Fig. 4, it uses selected network shapefiles, in particular those containing the data about the network geometry (like link, pocket lanes, etc.), and all shapefiles with snapshots taken for a given period of time. All these shapefiles are earlier created by means of ArcNet and ArcSnapshot programs.

However, since the number of snapshot shapefiles may be extremely high (for example 86400 for a 24-hour simulation with snapshots taken second by second), loading all of them at the startup may take a long time or even be impossible due to insufficient memory. Therefore, only network shapefiles are loaded into memory at the startup; vehicle snapshot shapefiles are loaded on the fly (one snapshot shapefile per one step of visualization).

TRANSIMS Visualizer was implemented in Java; thus it can be run on almost any computer, regardless of the installed operating system. For geospatial data manipulation and visualization, which is a key functionality of any GIS system, it uses an open source Java-based GeoTools library [4]. The current version of Visualizer enables the user to run the visualization forward or backward in a step by step or continuous mode. Beside a typical zooming functionality, it is also possible to hide/show selected layers and to customize their appearance. Due to the detailed vehicle data contained in the snapshot shapefiles, one may display various properties of each single vehicle and watch their changes throughout a simulation.

3.3. Animated movies

Beside the interactive visualization, another way to visualize a traffic flow simulation is to create an animated movie. This option was implemented as a two-step procedure shown in Fig. 5.

The first step consists in converting shapefiles into a sequence of images. Since each image represents a specific time of simulation, each of them is created from arbitrary selected network shapefiles and one shapefile containing the snapshot of that time. This step is done by the author's Shp2Img converter, which is written in Java and uses the GeoTools library for handling geospatial data.

Once the image sequence is created, it can be then converted into a movie by means of the FFmepeg software. FFmpeg is "a complete, cross-platform solution to record, convert and stream audio and video. It includes libavcodec - the leading audio/video codec library" [5].

One should note that the idea of the two-step procedure was originally proposed by Hubert Ley [6]. The above-presented schema is analogous to the original one,

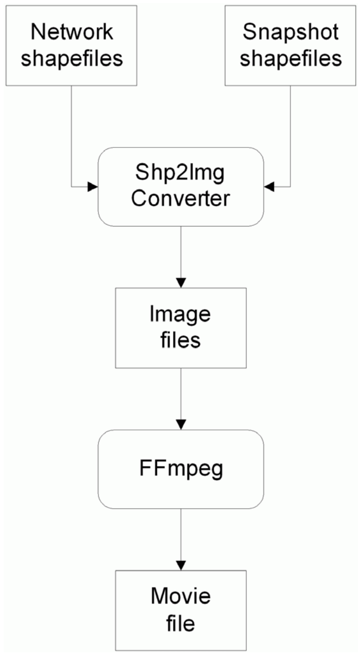

Fig. 5. Creation of an animated movie file
Source: [own work]

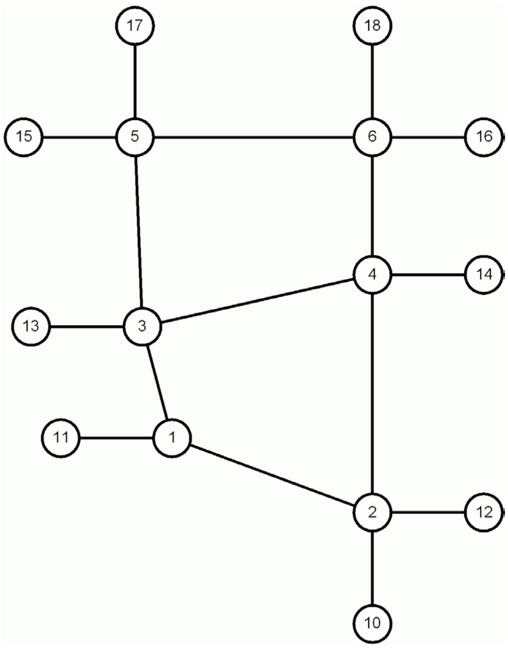

Fig. 6. Structure of the network
Source: [own work]

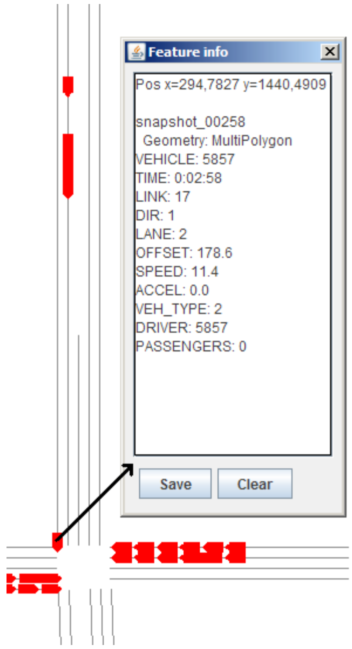

Fig. 7. "Feature info" window
Source: [own work]

except for converting shapefiles to images. In Ley's procedure the conversion is carried out by the shp2img converter from the FWTools kit [7]. However, due to some errors in shp2img, the converter very often fails.

Although animated movies, contrary to the interactive visualization, do not allow for any interaction, this form of visualization has also many advantages over the interactive one. First of all, movies are easy to distribute (just one file). Moreover, for playing movies only a standard movie player is required whereas Visualizer requires Java Runtime Environment to be installed on a computer.

When creating the visualization for large-scale scenarios, one may encounter problems with insufficient hardware capabilities. In particular, if computation power is very low, a process of creating a movie may take a long time. In order to speed up this process, one can run it parallel on a computer cluster or even on loosely-coupled computers. The first stage of the procedure can be easily parallelized, whereas parallelization of the second stage is somewhat harder. However, the most straightforward solution for this issue is to create a set of movies (one movie for each individual period of simulation time) instead of one.

4. Visualization example

In order to demonstrate the most essential capabilities of the Visualizer, an example of visualization of traffic in a segment of a real urban network is presented.

Let us consider a network presented in Fig. 6. The network is built of nodes (presented with identifiers) and two-way links. Nodes 1,…,6 are intersections, and 10,…,18 are boundary nodes, where each route starts and ends. The network model represents a part of a real road network in Poznan, Poland, and was used in simulation research conducted with TRANSIMS [8,9].

Fig.8 shows the main window of Visualizer. On the right side of the window, there is a view area presenting

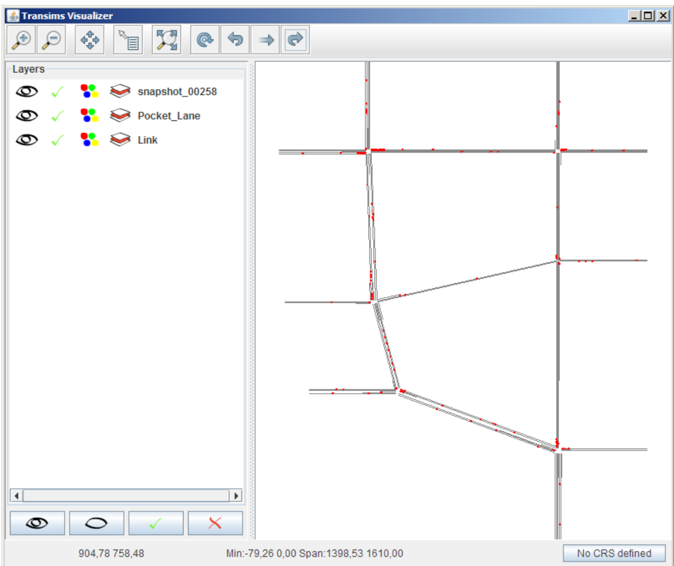

Fig. 8. Main window of the Visualizer
Source: [own work]

a course of simulation whereas on the left side there is a table containing a list of all loaded layers. In this example, three layers were loaded:

- snapshot_00258 (the top layer) – a snapshot of vehicles at time 0:02:58 (h:mm:ss); contains vehicle state details at that particular moment of simulation. For each time-step of simulation a relevant snapshot layer is loaded.
- Pocket_Lane (the middle layer) – contains additional lanes usually located at intersection inlets/outlets.
- Links (the bottom layer) – represents all links in the network.

As mentioned earlier, users can operate on layers. It is possible to add/remove or show/hide any layer. Additionally, one can change appearance of any layer. The zooming in/out functionality allows for taking a closer look at selected parts of the network. The "Feature info" option enables one to view properties of any element of any layer. For example, as shown in Fig. 7, the "Feature info" window contains detailed information about vehicle 5857 at time 0:02:58.

5. Conclusions

Visualization plays a crucial role in analyzing, calibrating and validating a traffic microsimulation. Due to the lack of official visualization software for TRANSIMS, one has to use third party tools in order to visualize simulation results. The presented tools, both the interactive visualizer and the movie creator, proved their usefulness during the author's research [8,9]. Moreover, the applications are written in Java and use open source libraries/programs (e.g., GeoTools, FFmpeg); hence, anyone can use them free of charge.

Bibliography:

[1] http://www.transims-opensource.net/
[2] WARDROP J. G., Some theoretical aspects of Road traffic research, In: Proceedings of the Institution of Civil Engineers, vol. 1 of 2, 1952.
[3] http://www.civil.utah.edu/~zhou/NEXTA_for_TRANSIMS.html
[4] http://www.geotools.org/
[5] http://www.ffmpeg.org/
[6] http://www.transims-opensource.net/index.php?option=com_agora&task=topic&id=474
[7] http://fwtools.maptools.org/
[8] MACIEJEWSKI M., MACIEJEWSKI M.: An application of cellular automata to simulation of urban traffic. Logistics 2/2010, pp. 1159-1168 (in Polish).
[9] MACIEJEWSKI M.: A comparison of microscopic traffic flow simulation systems for an urban area. 2nd International Scientific Conference "Transport Problems", Katowice-Kraków, Poland, June, 8-11, 2010 (in print).