

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/333402170>

A Smart Integrated Environment for Vehicular Traffic Simulation

Article in *International Journal of Communication Systems* · May 2019

CITATIONS
0

READS
43

6 authors, including:



Syed Waqar Jaffry

Punjab University College of Information Technology

55 PUBLICATIONS 229 CITATIONS

[SEE PROFILE](#)



Mian Mubasher

University of the Punjab

8 PUBLICATIONS 8 CITATIONS

[SEE PROFILE](#)



Muhammad Murtaza Yousaf

University of the Punjab

26 PUBLICATIONS 129 CITATIONS

[SEE PROFILE](#)



Dr Shahzad Sarwar

University of the Punjab

36 PUBLICATIONS 123 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



NLP for IOT [View project](#)



Image steganography [View project](#)

RESEARCH ARTICLE

A smart integrated environment for vehicular traffic simulation

Mian Muhammad Mubasher¹  | Syed Waqar Jaffry¹  | Muhammad Murtaza Yousaf¹  |
Imran Sarwar Bajwa²  | Shahzad Sarwar¹  | Laeeq Aslam¹ 

¹National Center of Artificial Intelligence, Punjab University College of Information Technology, University of the Punjab, Lahore, Pakistan

²Department of Computer Science and IT, The Islamia University Bahawalpur, Bahawalpur, Pakistan

Correspondence

Syed Waqar Jaffry, National Center of Artificial Intelligence, Punjab University College of Information Technology, University of the Punjab, Lahore, Pakistan.

Email: swjaffry@pucit.edu.pk

Summary

Globally, urbanization has increased due to the availability of the facilities, jobs, and better standard of living in cities. Increase in urban population has caused increase in urban commute. The increased pressure on urban settlements has made design and optimization of urban civil infrastructure very challenging especially road transport infrastructure. Domain experts such as transportation managers, engineers, scientist, and academics need effective, easy, and readily usable computational modeling and simulation tools to conduct systematic inquiry for design and optimization of the transportation system. Free and open-source software (FOSS) community has developed many vehicular traffic simulation software that are cost-effective and extensible, however are difficult to use. Usually, domain experts do not have hands-on software skills such as XML-based interfaces, software source versioning, and build management. To bridge this gap, in our work, an integrated development environment (IDE), namely, integrated vehicular traffic simulation environment (ITE), has been developed. ITE is developed on top of an existing FOSS multimodal open-source vehicular-traffic simulator (MovSim). ITE includes a scenario building tool for domain experts. It could be used to model road infrastructure and vehicular traffic in an integrated manner. The tool could be used to build and experiment novel traffic infrastructures and policies. Our work helps in reducing learning curve of domain experts by providing a free and open source, easy and readily usable vehicular traffic simulation tool.

KEYWORDS

computational modeling and simulation, integrated development environment, MovSim, road infrastructure modeling, vehicular traffic simulation

1 | INTRODUCTION

Civil engineering and its subdomains in general are very relevant disciplines because these disciplines affect sustainability and growth of today's world. Among the subdomains, transportation science, engineering, and management are very important due to their role in sustainable development, specifically road transportation, because it directly affects daily lives of general masses. Engagement of general masses in huge count makes road transportation very complex phenomenon.

Transportation is a complex phenomenon due to the number of participants, ie, drivers, pedestrians, etc. Complexity of this phenomenon is continuously growing as the population influx has increased in metropolitan due to availability of services and opportunities.¹ Around the world, tens of millions of people are living in couple of thousand-kilometer square in large cities. Huge population commutes on daily basis. Increased population of commuters and cost-effectiveness of transportation has played a major role in increase of vehicle count on roads. Due to sheer number of vehicles, complexity and challenges of vehicular traffic flow are also increasing. Due to this complexity, it is very difficult to analytically study and improve this sociotechnical system.

To study and analyze complex systems, computational science could be employed. Computational modeling and simulation is a branch of computational science that has several applications in diverse disciplines. It assists other disciplines to explore dynamics of complex systems by providing methods and tools to model and simulate relevant complex systems. Many disciplines, eg, physics, chemistry, biology, psychology, sociology, and economics, use computational modeling to understand, validate, experiment, and enhance the respective domain knowledge. Transportation is also a complex sociotechnical system; therefore, computational modeling and simulation is suitable to study this multidisciplinary domain.

Due to the vital role of transportation, computational modeling has widely been applied, and various aspects of the vehicular transportation have benefited through it.² The challenges in transportation science span over multiple dimensions including but not limited to optimal human resource deployment such as law enforcement personals, signal optimization, new infrastructure design and planning, effective use of existing infrastructure, avoidance of traffic congestion, jams, and accidents, and understanding of drivers' behavior. Computational modeling and simulation techniques applied to date try to address above mentioned problems through design, evaluation, and experimentation by simulation of computational models of vehicular traffic and transport infrastructure.

Simulation tools help the domain experts to simulate complex system, to learn insights. Here, simulation tools refer to a piece of software that provides implementation of computational models. Simulation tools are commonly referred as simulators. Simulators are used for policy design and analysis.³ Simulators could also be used in teaching and training.⁴ Currently, many simulators are available, which could be used to simulate vehicular traffic. Some of these simulators are proprietary, and others are free and open-source software (FOSS).⁵

In comparison with proprietary, FOSS is more cost-effective. Open-source development model is used to develop all types of software, which includes hardware drivers, operating systems, and applications software. Open-source software is more transparent with respect to execution and data security as source code remains available, which could be inspected. Relevant user-base derives the development of FOSS. Development and quality assurance of FOSS is more scalable in comparison with propriety software as most of the user-base act as development and quality assurance work force. FOSS also has user-base and development workforce from diversified backgrounds, which is an opportunity for innovative ideas itself. FOSS code-base remains available publicly; therefore, FOSS could always be adapted to specific needs.

There are many FOSS vehicular traffic simulators that are available to domain experts free of cost.⁶ But in many cases, these simulators do not provide easy-to-use interfaces to model vehicular traffic scenarios. These tools may be convenient for people from computing background, but domain experts may not be able to use them. It is a common practice that these simulators provide extensible mark-up language (XML)-based interfaces, which is a computer-friendly file format. Domain expert needs simulators to conduct what-if analysis, but usually the domain experts are not very convenient with computation and information technology, eg, XML. Hence, it is very much required that simulators must be readily accessible and conveniently usable for domain experts. Therefore, to reduce learning curve of simulator use and let domain experts focus on their core expertise, it is required that simulators must provide convenient graphical user interface (GUI) to model and simulate vehicular traffic. It is worth noticing that in this article, transportation academics, scientists, engineers, and managers are referred as domain experts.

To address above stated issues, in this work, state-of-the-art vehicular traffic simulators are explored, which are presented in Table 1. Commonly used proprietary and FOSS simulators have been identified. A suitable FOSS simulator named multimodel open-source vehicular-traffic simulator (MovSim) has been selected.⁷ Selection of MovSim is based on the number of reference computational models that have been implemented in it, use in different scientific studies, software modularity, and platform independence. To address above stated issues, a GUI-based integrated environment is developed on top of MovSim.

MovSim requires two XML files as input and provide simulation logs as result. One input file provides road logic information; this file follows OpenDRIVE specification. The other file provides information regarding vehicular traffic; this file follows xProject specification. MovSim also provides a visualization module to visualize simulation but does not

TABLE 1 Vehicular traffic simulators

Simulator	License
Vissim	Commercial
SUMO	FOSS
CORSIM	FOSS
Paramics	Commercial
SCATS	Commercial
UrbanSim	FOSS
ARCHISIM	FOSS
Aimsun	Commercial
MovSim	FOSS
MITSIM	FOSS

offer a GUI to design a traffic scenario. An effort has been made to fill this gap. An easy-to-use GUI tool has been developed on top of MovSim so that road network could be created and manipulated easily.⁸ The tool does not use standard OpenDRIVE specification; rather, it uses a modified version of OpenDRIVE. Another limitation is that it does not provide interface to express simulation configurations; therefore, the configurations are still required to be written in xProject XML specification manually. The tool is also not in public domain so far. To fix these issues, an easy-to-use GUI-enabled integrated environment, namely, integrated vehicular traffic simulation environment (ITE), has been developed on top of MovSim.

There is precedence in research community to develop effective GUI tools on top of exiting simulators. In the past, integrated environments with rich GUIs have been made on top of simulators to make the modeling and simulation process more efficient.⁹ Similarly, such advance tools have been developed for transportation modeling and simulation packages. For instance, PIMTRACTS is a simulator that has provision to simulate massive parallel large-scale simulations.¹⁰ In this simulator, special attention has been given to GUI components. In similar manner, a web-based, easy-to-use traffic simulation environment has been developed on top of an existing simulator, namely, Aimsun.¹¹⁻¹³ The tool is designed to teach various concepts of transportation engineering to undergraduate students with the help of computer simulations. In another similar effort, a simulation tool has been developed to teach road geometry to transportation engineering students.¹⁴ In another study, a web-based simulation and visual analytics tool has been developed that helps domain experts in conducting different simulation-based studies.¹⁵

MovSim provides rich set of reference microscopic models. The tool is also complimented with rich support of literature, but it is not readily available and easy to use for users who are not familiar with versioning control systems, source building systems, and XML technology. To solve these issues, efforts have been made, but there are few limitations in existing efforts that have been discussed in detail, in Section 2.

The remainder of this article is organized as follows. Section 2 presents background that highlights prominent works, their contribution, and importance in the field of transportation science and engineering and gaps in state-of-the-art. Section 2 also emphasizes the need of the proposed tool. After that, functional requirements, general workflow, software architecture, demonstration and training resources, and software distribution of ITE has been described in Section 3, which is followed by the conclusion and future work in Section 4.

2 | BACKGROUND

This section describes existing efforts that have been made so far to develop an easy-to-use, readily available traffic simulation environment. MovSim has been selected as core simulation engine because it is free and open-source tool; it is modular and platform independent. MovSim expects two input files expressed in OpenDRIVE and xProject specifications, respectively. An effort has already been made to replace this XML-based input mechanism with GUI,⁸ which would be referred as EasyMovSim in following sections. All the related background work and existing gaps have been discussed at length in following five subsections, namely, OpenDRIVE, xProject, MovSim, EasyMovSim, and Gaps in state-of-the-art.

2.1 | OpenDRIVE

OpenDRIVE is an open and cross-vendor specification to represent road network logic information.^{16,17} It is a machine-friendly XML file format having a specific file extension, ie, XODR. It was initially released in 2005 with version 0.7, and current release was distributed in 2015 with version 1.4. The premise was to have a standard file format to express road logic such as geometry, lanes, junctions, signals, and relationships among them so that transportation-related computer application that requires road network information could be developed on top of it.

In a XML file, individual roads are defined in OpenDRIVE. Each road definition carries information about its geometry, lanes, traffic source, and traffic sink. OpenDRIVE also carries information about junctions and traffic controllers. It uses a simple element hierarchy to describe the road network. OpenDRIVE road network specification has been used to develop many applications such as

1. Prototyping of advance driving assistance system¹⁸
2. Integration of road network logic in virtual reality¹⁹
3. Conversion of measured data from real road to OpenDRIVE data to develop virtual models of real roads^{20,21}
4. Fusion of alternative data sources such as cadastral or mobile mapping data to generate OpenDRIVE data of real road network²²

OpenDRIVE provides a standard representation mechanism for information related to roads to develop transportation-related computer applications. MovSim also uses OpenDRIVE specification to represent road logic information to simulate traffic.

2.2 | xProject

xProject is a MovSim-specific file format. It is an XML file that uses a specific file extension, ie, XPRJ. It carries information about traffic scenario for simulation. Generally, a traffic scenario is composed of road infrastructure, which is defined in a separate OpenDRIVE file, and description of vehicular traffic, which is desired to be simulated. The xProject file carries information about vehicle prototypes that would be simulated on the road network. Vehicle prototypes carry information about physical properties of vehicle. It also carries information about car-following and lane change models along with their parameters to express vehicle's movement dynamics.

Using traffic prototypes or types of vehicle, traffic compositions are defined in xProject file. A traffic composition describes ratio of each vehicle prototype that would be spawned in simulation and number of vehicle per unit time. Once a traffic composition is defined, it could be attached to a road as traffic source. In complex road networks, a vehicle can opt from a set of routes. Routes are series of connected roads section. A traffic composition could be assigned a specific route to follow. Phases of traffic lights are also defined in xProject file. The file also carries output configurations. Output configuration tells simulator which information must be logged or recorded about the simulation for analysis purposes.

2.3 | MovSim

MovSim is used to conduct simulation-based experiments related to transportation science and engineering.⁷ MovSim is an open-source, modular, platform-independent (Java-based) software. The project was first publicly distributed in late 1990s as an Internet applet at the website <http://www.traffic-simulation.de>. The project has also been presented at German computer exhibition CeBIT in 2009 and 2010. The project got coverage in media such as Wall Street Journal (July 1, 2005) due to the simulator's application in verifying and validating infrastructure, technology, and policy innovations. There are two major modules of MovSim. One module is responsible for simulation, and the other is responsible for graphical visualization of the simulation. Simulation engine is named as core, and visualization module is named as viewer. The core can simulate a vehicular traffic scenario, but it does not provide visualization of it; rather, it logs simulation results as comma separated values (CSV) in text files for analysis purposes. However, visualizer helps in understanding the traffic scenario by providing visualization of traffic on provided road infrastructure. A screenshot of the visualizer is presented in Figure 1.

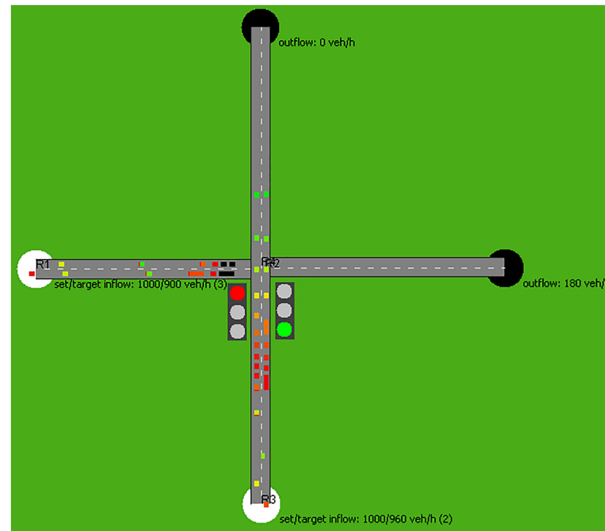


FIGURE 1 MovSim viewer/visualizer

MovSim provides implementation of many reference microscopic vehicular traffic models. These models can be categorized further into two classes such as car-following models or longitudinal models and lane change models or lateral movement models. Car-following models simulate longitudinal dynamics such as acceleration and breaking whereas lane change models simulate lane selection decision making of an individual vehicle.

MovSim provides implementation of reference time continuous car-following models such as intelligent driver model (IDM),²³ optimal velocity model (OVM),²⁴ and full velocity difference model (FVDM).²⁵ It also provides implementation of several iterated graph models such as Gipps model²⁶ and Krauss model.²⁷ The simulator also provides reference implementation of cellular automata models such as Nagel-Schreckenbreg (NaSch) model²⁸ and Kerner-Klenov-Wolf (KKW) model.²⁹ Along with car-following models, MovSim also provides reference implementation of microscopic lane change model, namely, MOBIL. MOBIL could be used to compliment exiting microscopic car-following models for multilane vehicular traffic simulation.³⁰ MovSim also have implementation of driver's memory model and acceleration noise. MovSim could be used to study travel time, fuel consumption, and spatiotemporal contours with the help of microscopic data of individual microscopic entities, ie, vehicles. It may also collect data from simulated on-road cross-sectional detectors. MovSim also provides implementation of a physics-based microscopic fuel consumption model. It uses parameters such as speed, acceleration, gear, and mass of the individual vehicle. The fuel consumption model also keeps into account road friction, road gradient, and wind drag to estimate fuel consumption of individual vehicle.

In recent past, MovSim has been used for a variety of research studies. MovSim has been used as a core vehicular traffic simulation engine in a smartphone game to crowdsource signal optimization problem.³¹ In another study, a marathon competition has been simulated with the help of MovSim.³² MovSim has been used for simulation of autonomous vehicle agents enabled with a multiobjective method to find multiple prioritizations for passengers traveling in an autonomous vehicle so that the passenger may chose the way car should be drove.^{33,34} To incorporate human factors in microscopic driver models, a model of personality profile has been incorporated in existing microscopic models, and simulation-based experiments have been performed using MovSim.³⁵

Enhancements have also been made in MovSim core. For instance, MovSim has been extended to simulate vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-road (V2R) communication so that vehicular ad hoc network (VANET)-related simulation experiments could be performed.³⁶⁻³⁹ To make traffic simulation more realistic, a sequential Monte Carlo method has been devised for event reconstruction and real-time data assimilation into MovSim.^{40,41} In other studies, particle filtration methods have also been incorporated in MovSim.^{42,43} Apart of the fact that MovSim has been used in recent past in broad range of research studies, a major plus point of MovSim is to provide implementation of reference microscopic models.⁴⁴ Hence, MovSim is a very good candidate to be used in computational transportation research, but it lacks an easy GUI to design traffic scenarios.

MovSim has easy-to-comprehend modular software architecture; hence, it is easy for computer science experts to understand and extend MovSim as per requirements. To simulate a traffic scenario, MovSim requires two XML-based input files, ie, OpenDRIVE for road network and xProject for vehicular traffic. These files are written as per prescribed

specification or schema in common text editing tools. Once OpenDRIVE and xProject files are ready, MovSim could be used using command line interface (CLI) to simulate desired traffic scenario. A range of different traffic scenarios are built-in and can be simulated using MovSim, following list states few applications of MovSim.

1. Study of interesting traffic scenarios, eg, traffic congestion, jam, and accidents
2. Study on different road infrastructures, eg, roundabouts, junctions, underpasses, and bridges
3. Simulation based experiments of vehicles with special equipment, eg, adaptive cruise control
4. Simulation based experiments to study VANET, eg, V2V and V2I communication
5. Simulation-based experiment to estimate fuel consumption and emission
6. To be used as traffic simulation engine in other applications such as computer games

2.4 | EasyMovSim

An effort has already been made to provide a GUI on top of MovSim.⁸ The GUI-enabled software is referred as EasyMovSim in this article. In EasyMovSim, changes have been made in standard OpenDRIVE specification. As per changes in OpenDRIVE, MovSim core has been updated to accommodate the changes. The changes in OpenDRIVE specifications are made to attach multiple sources and sinks to a lane. The change required updates in default implementation of junction in MovSim as well. A junction attaches multiple roads and consequently lanes to each other. In default implementation, one lane can attach to only one lane as source or sink; however, in EasyMovSim implementation, there can be many sources as well as many sinks. In upgraded implementation, selection of routes has also been upgraded to cater changes in OpenDRIVE. Concept of critical sections on intersections has also been introduced, such as if there are junctions where congestion and accidents can occur, acceptable speeds of the simulated vehicles are overridden by safe speed limits to avoid undesirable situations such as congestion, jam, and accidents.

EasyMovSim provides a GUI to draw road network. The GUI uses Google Street View service to provide a background image to show a real-world road network. The background could be used as a drawing trace paper since road network could be manually drawn on top of the roads shown in selected background. Authors have stated their intention to incorporate Open Street Map (OSM) in the future. OSM is an open cartographic database, which provides its own XML-based structure called OSM. OSM specification could be used to automatically generate road network logic in OpenDRIVE specification. Apart of road network drawing, the GUI of the tool provides a road network editor. The road network editor allows adding junctions with predefined commonly used junction templates such as T, Y, and + shape junctions. The road network editor also provides a traffic signal editor using which location, timing, and phases of signal could be defined.

2.5 | Gaps in state-of-the-art

MovSim could be used to simulate a vehicular traffic scenario. MovSim requires two XML-based files as input, ie, OpenDRIVE file for road network and xProject file for vehicular traffic. These XML-based input interfaces are convenient for computer science experts but are difficult for domain experts, eg, transportation academics, researchers, engineers, managers, and students. It would be significant contribution if this gap could be bridged. To fill this gap, an integrated traffic scenario development and simulation tool is required. Secondly, it is not trivial to build executable file from source code to use MovSim to conduct simulation-based experiments. The executable should be readily available for domain experts.

To fill above mentioned gaps, an effort has already been made, namely, EasyMovSim. But EasyMovSim is in neither public domain nor freely available. It uses a customized, nonstandard OpenDRIVE specification that is subjected to more research and investigation whether these changes should be made in OpenDRIVE specification or not. EasyMovSim provides support of drawing road network, which in turn helps generating OpenDRIVE file, but it does not use full potential of xProject format specification as most of the components in xProject file are still required to be defined manually by writing in xProject file using a text editor. To do so, one should have knowledge of OpenDRIVE and xProject XML specification. To cater above mentioned issues, in current work, an integrated environment for vehicular traffic simulation has been developed.

3 | INTEGRATED VEHICULAR TRAFFIC SIMULATION ENVIRONMENT

ITE is a free and open-source, java-based, platform-independent software. It is an easy-to-use and readily available tool. Domain experts can simulate vehicular traffic scenarios using ITE without being exposed to steep learning curve of XML-based OpenDRIVE and xProject specifications. In the following subsection, functional requirements, basic work flow, GUI, software architecture, training and demonstration resources, and software distribution of ITE has been discussed.

3.1 | Functional requirements

The purpose of this integrated environment is to provide a framework to domain experts where they may design scenarios to simulate road infrastructure and vehicular traffic. Considering the intended user-base, ITE has following functional requirements.

1. Interface to model road infrastructure
 - a. Geometry of roads
 - b. Connectivity of roads
 - c. Junctions
2. Interface to model vehicular traffic
 - a. Physical properties
 - b. Car-following and lane change model
 - c. Parameters of the models
3. Interface to model traffic controllers and policies
4. Interface to model traffic sources
 - a. Probability distribution of defined vehicle prototypes
5. Interface to define routes
6. Basic workbench utilities such as saving and loading traffic scenarios

These primitive functional requirements would help domain experts to design and simulate traffic scenarios using reference microscopic vehicular traffic models without being exposed to computing-related learning curve.

3.2 | Basic work flow

ITE provides a GUI to perform basic create, read, update, and delete (CRUD) operation to design a traffic scenario. A user may perform CRUD on roads, lanes, junctions, signals, vehicle prototype, routes, etc. While user designs a traffic scenario, ITE automatically generates appropriate OpenDRIVE and xProject files. To make sure that auto-generated OpenDRIVE and xProject files are semantically correct, it uses the flow presented in Figure 2.

When a user performs a CRUD operation on GUI, an appropriate event handler is invoked. The handler makes changes in OpenDRIVE and xProject files accordingly. In case the CRUD operation is invalid and may cause semantic errors in either OpenDRIVE or xProject file, an exception is thrown, and user is intimated about the error through specific alerts.

3.3 | Graphical user interface

This section explains GUI of ITE. Broadly, the GUI has five major components, namely, Tool Bar, Tool Panel, Drawing Area, Road Context, and MovSim Config Context. Tool Bar provides functionalities to load, save and simulate a traffic scenario. Tool Panel provides functionalities to add or remove roads, add traffic source to a road, and add or remove links between lanes on Drawing Area and consequently in a traffic scenario. Road Context provides functionalities to

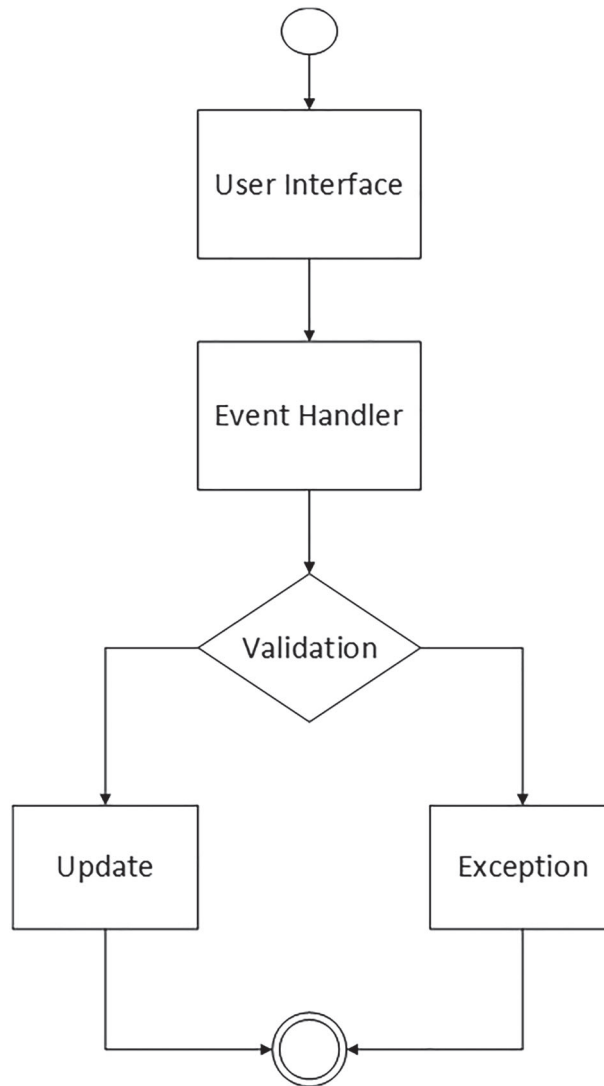


FIGURE 2 Basic work flow

change properties of a single selected road and junction. MovSim Config Context provides functionalities to configure general simulation-related properties, vehicle prototypes, traffic signal phases, routes, and simulation output. Main GUI of ITE has been presented in Figure 3.

Tool Bar, Tool Panel and Drawing Area are simple interfaces. Tool Bar has only three functions, and it may open an existing traffic scenario and save current scenario, and it may simulate current scenario. Tool Panel has five functions. It has a simple pointer, which may be used to select an item on Drawing Area to change its properties. It has functionalities to add a straight or curve road to the Drawing Area. It also has functionalities to add or remove a link between two lanes. Road Context and MovSim Config Context are relatively complex interfaces. The following two sections explain both GUI components in detail.

3.3.1 | Road Context

Road Context is further divided in two components, namely, Road Panel and Junction Panel. Road Panel and Junction Panel are presented in Figures 4 and 5, respectively. Once a road is drawn on Drawing Area, its name and description could be provided using Road Panel. Road Panel also provides interface to assign geometrical properties to a selected road, such as its coordinates, direction, length, and type (line or curve). Road Panel may also be used to add or remove lanes from a road. Once a lane is added, its width, type (driving, entry, or exit), and maximum speed could be provided. Signals are also added to a road on a specified position using Road Panel. OpenDRIVE specification allows to assign one

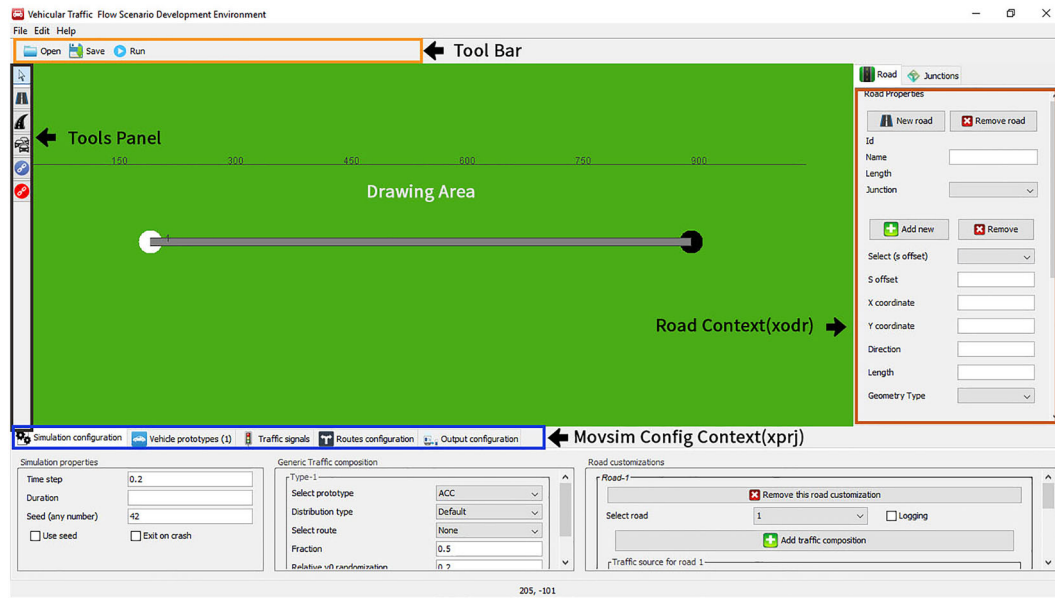


FIGURE 3 GUI of ITE

traffic source and one traffic sink to a road. However, in real road networks, a road can split into multiple roads, and multiple roads can merge into a single road. To address this issue, OpenDRIVE specification provides a special component, namely, junction. A junction does not occupy real estate in a traffic scenario; rather, it is a logical or virtual entity that handles splitting and merging of roads. Junction Panel provides a simple but effective interface to define junctions in a traffic scenario. A junction may join many roads with each other as source or sink. It is worth noticing that, still, a lane could only be attached with a single lane as source or sink. A junction may compose of multiple connection. To establish a connection, it needs connecting road and incoming road. It also needs a type of connection point (start or end). Once two roads are connected via junction, their lanes are mapped by providing “from” and “to” lanes by mentioning their ids.

3.3.2 | MovSim Config Context

MovSim Config Context is composed of five panels, ie, Simulation Configuration Panel, Vehicle Prototype Panel, Traffic Signals Panel, Routes Configuration Panel, and Output Configuration Panel. Each panel provides interface to express information about specific aspects of traffic scenario. The following paragraphs briefly discuss each panel.

Simulation Configuration Panel provides three interfaces, ie, Simulation Properties Panel, Generic Traffic Composition Panel, and Road Customization Panel. These interfaces are presented in Figures 6–8, respectively. Simulation properties panel is mainly composed of time step of simulation, duration of simulation, and a random number to be used as simulation seed. Generic Traffic Composition Panel provides interface to define traffic composition, which is assigned to every traffic source by default. However, if it is required to assign a different traffic composition to a traffic source, Road Customization Panel could be used to assign a custom traffic composition to a road as a specific traffic source.

To define generic and specific or custom traffic compositions, at least one vehicle prototype is required. When a traffic scenario is made in ITE, a vehicle prototype is automatically made and assigned to a generic traffic composition. Vehicle Prototype Panel provides interface to define vehicle prototype. A vehicle prototype is composed of three major components, ie, physical properties, car-following model, and lane change model of vehicle. Physical properties of a vehicle include length, width, and maximum deceleration, which could be assigned through Properties Panel. Driving Models Panel provides option to select a reference car-following model and allows a user to provide values of model parameters. Whereas only one lane change model MOBIL is available in MovSim, Lane Change Model Panel allows user to assign parameters of MOBIL model. It is worth noticing that assigning a lane change model to a vehicle prototype is optional; however, if not assigned, the vehicles generated using the prototype will never change their lane. Figures 9–11 present interfaces that allows user to assign physical properties, car-following model, and lane change model to a vehicle prototype, respectively. The configuration details of signal controllers are depicted in Figures 12 and 13.

Road **Junctions**

Road Properties

New road **Remove road**

Id: 1
Name: R1
Length: 800.0
Junction: None

Geometry

Add new **Remove**

Select (s offset): 0.0
S offset: 0
X coordinate: 40
Y coordinate: 300
Direction: 0
Length: 800
Geometry Type: line

Lanes (2)

Add new **Remove**

Lane width: 10.0
Select lane (Id): -1
Type: driving
Level: 0
Max speed:

Signals

New signal

Link

Successor:
Element id: 2
Element type: road
Contact Point: start

FIGURE 4 Road Panel

The image shows a software interface titled "Junctions". At the top, there is a "Select junction" field with the value "1009". Below this are two buttons: a green "+" button labeled "New junction" and a red "X" button labeled "Remove".

The main area contains two sections for configuring connections:

- Connection id: 0**: Includes a "Remove connection" button (red X), "Con. Rd" set to 2, "In. Rd" set to 4, and "Cont. point" set to start. Below these are "From" and "To" dropdowns with values -3 and -1, and a "New lane link" button (green +).
- Connection id: 1**: Includes a "Remove connection" button (red X), "Con. Rd" set to 2, "In. Rd" set to 3, and "Cont. point" set to start. Below these are two pairs of "From" and "To" dropdowns with values (-2, -2) and (-1, -1), and a "New lane link" button (green +).

At the bottom of the panel is a "New connection" button (green +).

FIGURE 5 Junction Panel

Routes Configuration Panel provides interface to define a sequence of connected roads as a route. A route could be assigned to a traffic composition. A route is an optional parameter of traffic composition; if a route is assigned to a traffic composition, vehicles generated by that traffic composition strictly follow the specified route. If a route is not assigned to a composition, the generated vehicles just follow the proceeding driving road. Interface of route configuration has been presented in Figure 14.

Visualization of the simulation helps to understand vehicular traffic intuitively. However, quantitative analysis is not possible without obtaining simulation logs or data. Output Configuration Panel allows user to configure ITE about the type of data that must be logged during the simulation so that further analysis could be performed. Two types of data could be logged, namely, floating car data and travel time data. Floating car data provide microscopic information about individual vehicle whereas travel time data provide macroscopic data of traffic on a given route. Both types of data logs are generated in comma separated values (CSV) format. Data logging of specific vehicles could be configured using

Simulation properties

Time step	<input type="text" value="0.2"/>
Duration	<input type="text"/>
Seed (any number)	<input type="text" value="42"/>
<input type="checkbox"/> Use seed	<input type="checkbox"/> Exit on crash

FIGURE 6 Simulation Properties Panel

Generic Traffic composition

Type-1

Select prototype	ACC1
Distribution type	Default
Select route	east
Fraction	<input type="text" value="0.8"/>
Relative v0 randomization	<input type="text" value="0.2"/>

Type-2

Select prototype	ACC2
Distribution type	Default
Select route	south
Fraction	<input type="text" value="0.2"/>
Relative v0 randomization	<input type="text" value="0.2"/>

FIGURE 7 Generic Traffic Composition Panel

Road customizations

Road-1

Select road: Logging

Traffic source for road 1

Logging

Inflow-1

T	<input type="text" value="0.0"/>
Quantity/hour	<input type="text" value="1200.0"/>
V	<input type="text"/>

FIGURE 8 Road Customizations Panel

Properties

Prototype name	<input type="text" value="ACC"/>
<i>Vehicle specifications:</i>	
Vehicle length	<input type="text" value="6.0"/>
Vehicle width	<input type="text"/>
Max. Deceleration	<input type="text" value="9.0"/>

FIGURE 9 Properties Panel

Driving models

IDM

a	<input type="text" value="1.0"/>
b	<input type="text" value="1.5"/>
T	<input type="text" value="1.0"/>
Delta	<input type="text"/>
v0	<input type="text" value="0.0"/>
s0	<input type="text" value="2.0"/>
s1	<input type="text"/>

FIGURE 10 Driving Models Panel

Lane change model

Safe deceleration	<input type="text" value="4.0"/>
Minimum gap	<input type="text" value="2.0"/>
Threshold acceleration	<input type="text" value="0.2"/>
Right bias acceleration	<input type="text" value="0.3"/>
Politeness	<input type="text" value="0.1"/>

FIGURE 11 Lane Change Model Panel

Signal Controllers

Controller

Name

Signal

Signal-593

Move to controller

FIGURE 12 Signal Controllers Panel

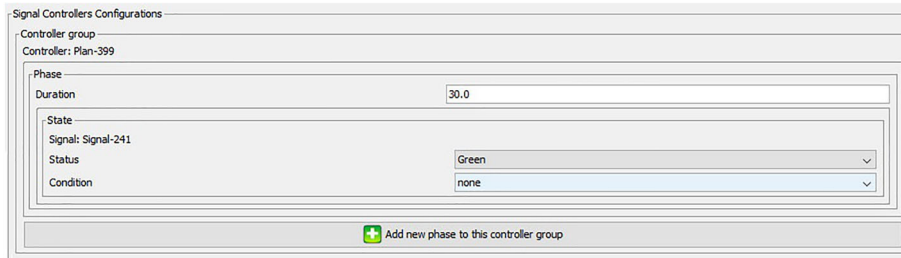


FIGURE 13 Signal Controllers Configurations Panel

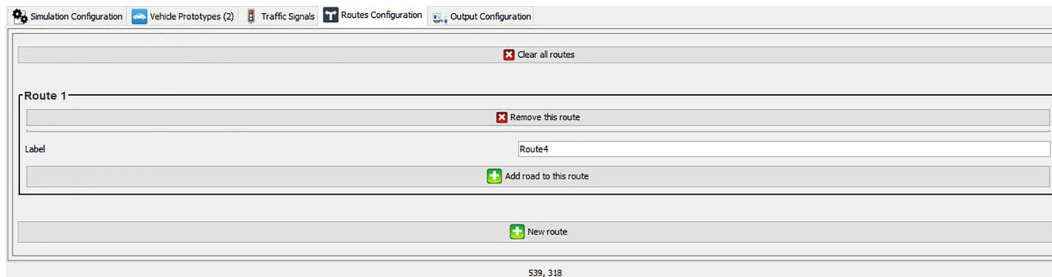


FIGURE 14 Route Configuration Panel

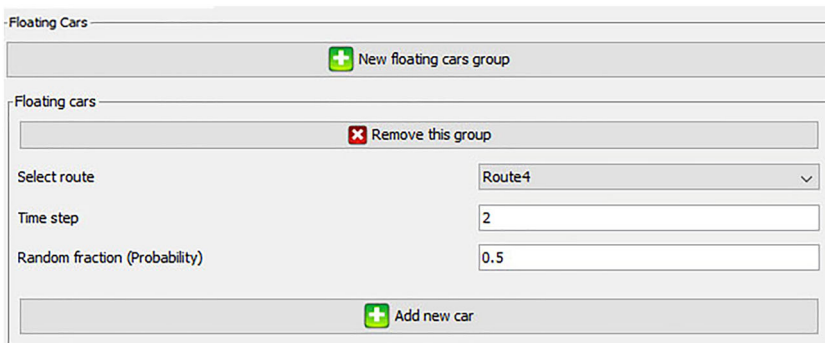


FIGURE 15 Floating Cars Panel

Floating Car Data Panel. Floating Car Data Panel is presented in Figure 15. Data logging (macroscopic) of traffic on a given route could be configured using Travel Time Panel. Travel Time Panel is presented in Figure 16. It is worth noticing that the simulation logs are generated in the same folder where executable of the simulator is placed.

3.4 | Software architecture

MovSim has a very simple and extensible modular software architecture. ITE has also been developed keeping same philosophy in view. ITE has clear and well-defined modules. It has five major software packages, ie, GUI, Event Handler, Validator, Updater, and Exception modules. GUI is further subdivided in App Frame, Tool Bar, Tool Panel, Drawing Area, Road Context, and MovSim Config Context. Figure 17 presents the software architecture of ITE. In the following subsections, important modules of ITE have been discussed. In Figure 17, each of the major component is represented by a rectangle, whereas the arrows show the association between the components such as a component on the head side of the arrow consumes service from component on the tail side.

3.4.1 | Graphical user interface

This section describes software architecture of ITE GUI. The ITE GUI has App Frame, Tool Bar, Tool Panel, Drawing Area, Road Context, and MovSim Config Context as major components. App Frame is the main container that holds all

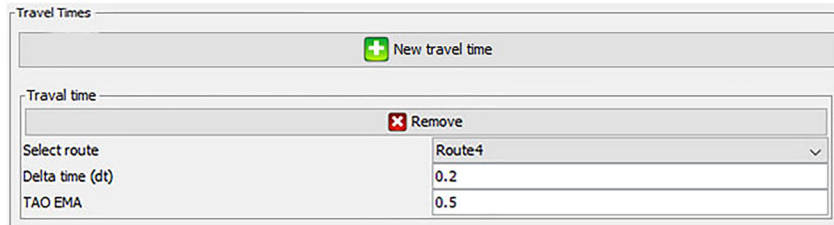


FIGURE 16 Travel Times Panel

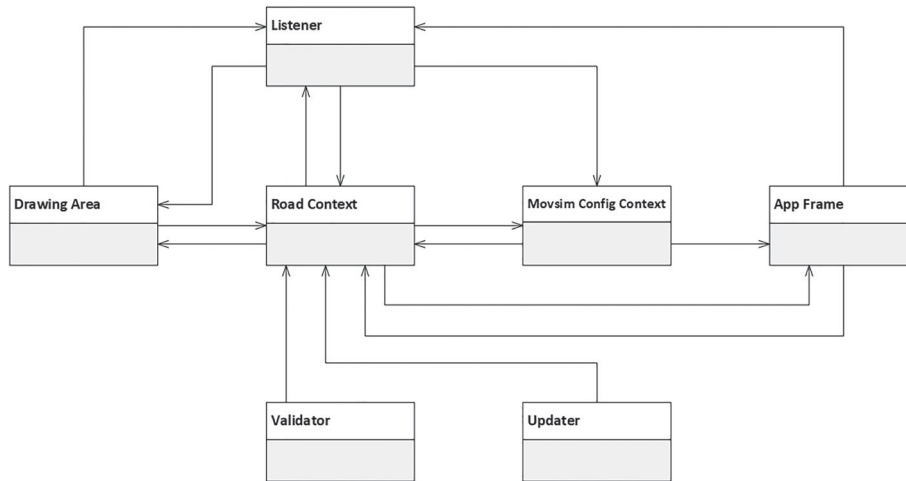


FIGURE 17 Software architecture

the GUI components and display them on screen. Tool Bar is composed of buttons to load, save, and simulate a traffic scenario. Tool Panel provides buttons to select objects on the Drawing Area. It also provides functionalities to draw new straight or arc roads and to attach or detach roads on Drawing Area. Drawing Area is a canvas object on which traffic scenarios are designed by drawing roads and creating connections between them. Road Context provides user interface to describe properties of a single road or junction. MovSim Config Context provides interface to describe simulation-related properties, vehicles, routes, and output of the simulation. Road Context and MovSim Config Context are each composed of several panels.

Road context is primarily composed of Road Panel and Junction Panel. MovSim Config Context is composed of Simulation Panel, Vehicle Prototype Panel, Traffic Panel, Routes Panel, and Output Panel. Graphical representation of GUI architecture has been presented in Figure 18.

3.4.2 | Event handlers

When a user performs a CRUD operation on GUI to make changes in the traffic scenario, an appropriate event is invoked. Corresponding event handler processes the invoked event. For every GUI component, there is an event listener or a set of event listeners that responds to interaction of the user. In Figure 19, all events listeners for every GUI component have been presented.

Although event handlers or event listeners are in a single software module or package and are peers in hierarchy, however, there is a logical grouping of these event listeners. The logical grouping is based on the UI components that are handled by these event listeners. Figure 19 depicts this logical grouping among the listener. In the following subsections, each logical group has been discussed.

App Frame

App Frame Listener handles user interaction of App Frame Menu, Tool Panel, and Tool Bar. These interactions are simple; hence, their handling has been performed by a single listener named App Frame Listener.

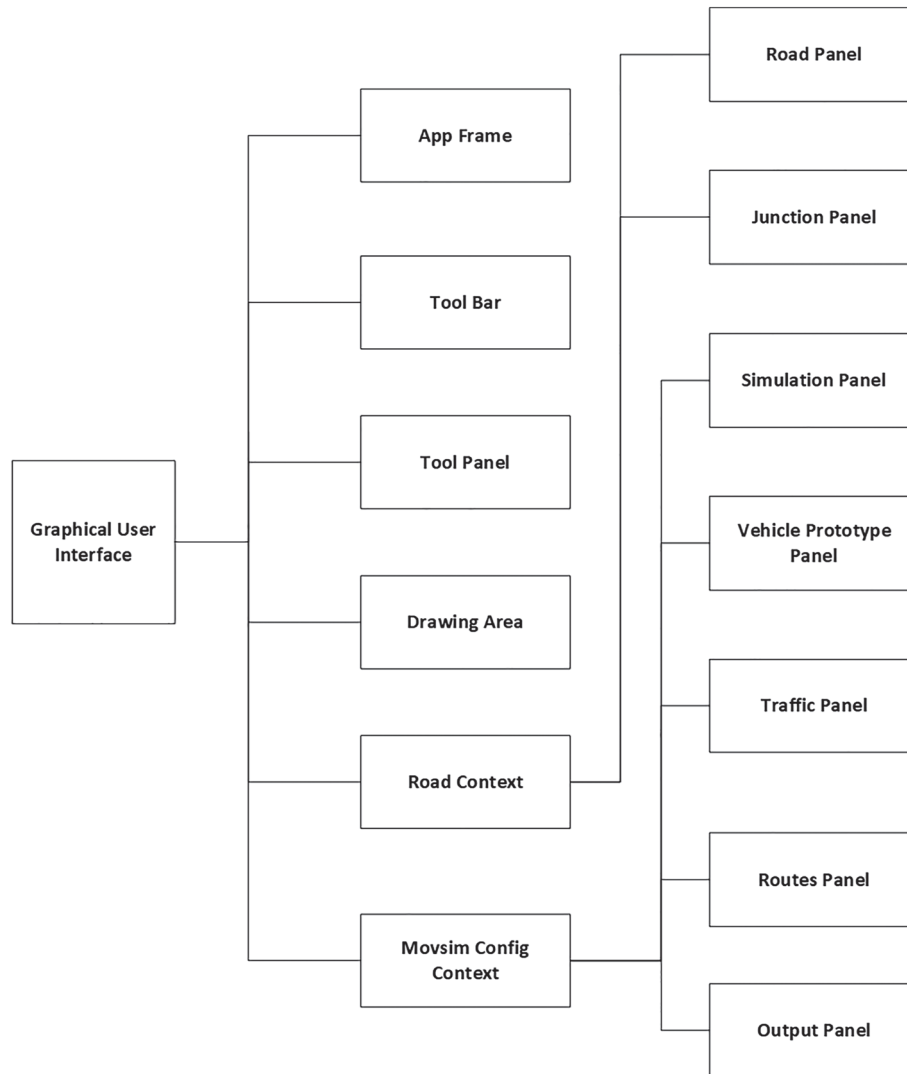


FIGURE 18 Architecture of graphical user interface

Drawing Area

Connection Listener, Lane Link Listener, Drawing Area Mouse Listener, and Drawing Area Popup Listener handle user interaction that occurs on Drawing Area. Connection Listener is invoked when user connects two roads on Drawing Area. Lane Link Listener is invoked when user attaches or detaches link between two lanes. Drawing Area Mouse Listener is involved when a user selects or displace objects on Drawing Area. It also handles zoom in and zoom out operation of Drawing Area. Drawing Area Popup Listener handles operation provided against right click on Drawing Area. These operations include appending a road section to an existing road, rotation of road, addition and deletion of lanes, and changing geometry of road such as curvature, direction, width, and starting point.

Road Panel

Road Panel is part of Road Context user interface. Geometry Panel Listener, Lanes Panel Listener, Link Panel Listener, Signals Panel Listener, and Signal Listener handle user interaction on Road Panel. Road Panel is composed of Geometry Panel, Lane Panel, Lane Link Panel, and Signal Panel. Geometry Panel allows to update geometry information of a road. Geometry Panel Listener handles user interaction on Geometry Panel. Lane Panel is used to configure lane settings such as count and width of lanes. Lanes Panel Listener handles user interaction with Lanes Panel. User may create and modify links between exiting lanes using Lane Link Panel. Link Panel Listener handles user interaction with Lane Link Panel. Signals could be attached to a road using Signal Panel. Signals Panel Listener handles user interaction on Signal Panel. Once a signal is attached to road, its properties such as location on road could be changed. Signal Listener handles user interaction with an individual signal.

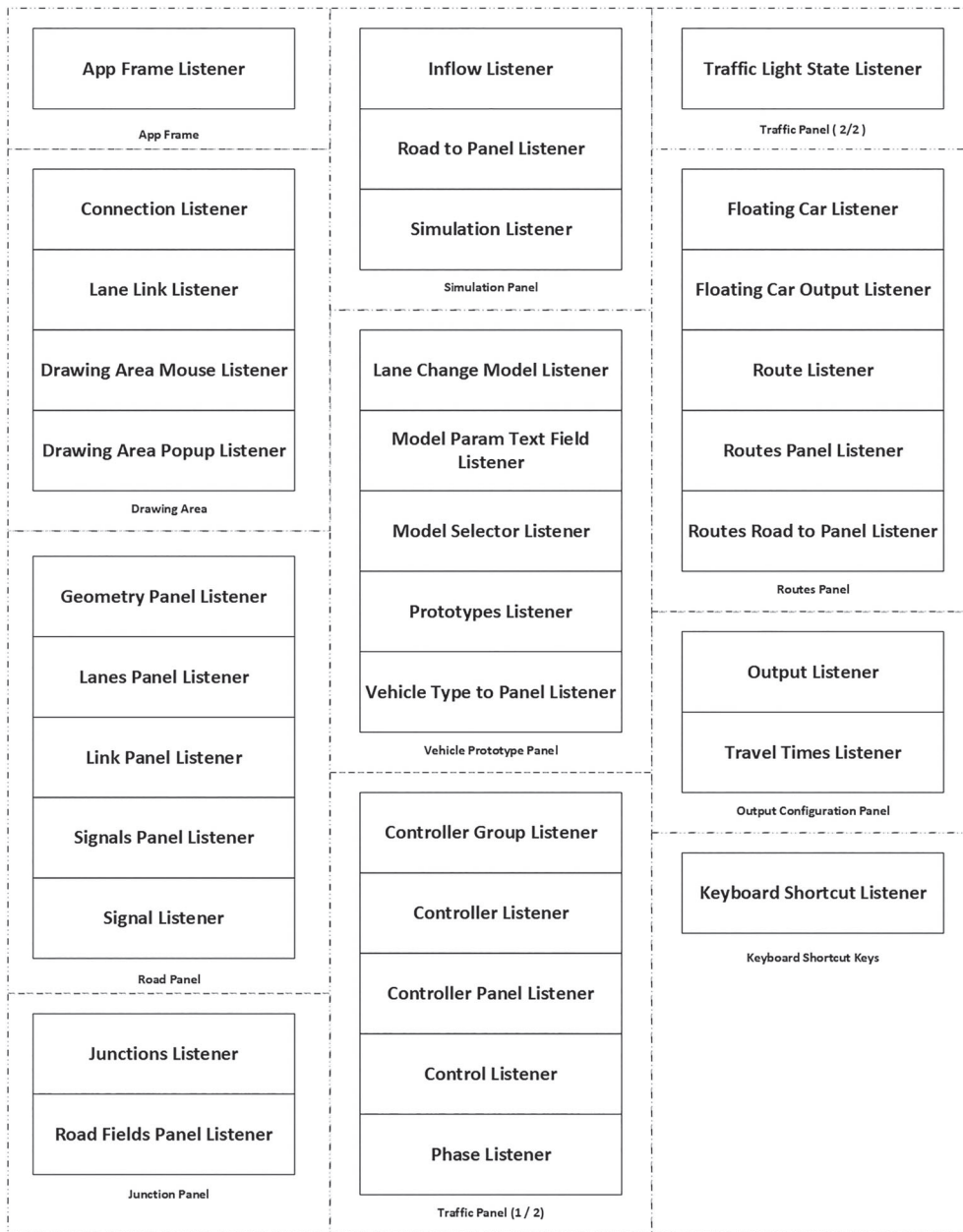


FIGURE 19 Architecture of event listener

Junction Panel

Junction Panel is part of Road Context. Junctions could be introduced in a traffic scenario using Junction Panel. Junctions Listener handles user interaction to create and delete junctions using Junction Panel. User may attach or remove roads from individual junction. Road Fields Panel Listener handles such events.

Simulation Panel

Individual roads may be assigned custom configuration such as inflow of traffic. User may provide this configuration; these configurations are provided on Road Customization Panel, which is part of Simulation Panel. Inflow Listener handles input given to customize inflow rates of individual roads, whereas Road to Panel Listener handles customization of other parameters on individual road level. Simulation Listener handles general settings of Simulation Panel.

Vehicle Prototype Panel

Vehicle Type to Panel Listener is triggered when a new vehicle prototype is added or removed. Prototypes Listener is triggered when an individual vehicle prototype is modified. Other than some general settings such as physical properties

of vehicle, users need to provide car following model, its parameters, and parameters of lane change model. In this regard, Model Selector Listener handles selection of car-following model. Lane Change Model Listener handles configurations of lane change model that are set using Vehicle Prototype Panel. Model Param Text Field Listener handles events when parameters of car-following models are defined.

Traffic Panel

Controller Group Listener handles definition of controller group of signals. Controller Listener is triggered on adding and removing a signal into/from a group. Controller Panel Listener is invoked when a user adds or removes a traffic light phase into a signal. Control Listener handles configuration of an individual controller. Phase Listener handles phase adjustment of a controller group. Traffic Light State Listener handles configuration of possible traffic light states.

Routes Panel

Route Listener handles definition of routes. Routes Panel Listener handles configuration of individual routes. Routes Road to Panel Listener is triggered upon adding new roads to a route; the listener checks if route remains valid (connected) after adding the road.

Output Panel

Output Listener handles output configuration of a simulation. Travel Time Listener handles output configuration related to travel time output that provides macroscopic data of simulated traffic. Floating Car Listener is invoked upon removal of a vehicle from cars that would be logged. Floating Car Output Listener handles definition of a floating car group, and it also handles definition of travel time logging configuration that provides information of all microscopic entities individually.

Keyboard Shortcut Keys

Keyboard Shortcut Listener handles keyboard shortcuts that have been implemented so that user may execute common operations such as to save a scenario (CTRL + S) and to undo an unintended change in a scenario (CTRL + Z) easily using keyboard shortcuts.

3.4.3 | Validators

When user perform a CRUD operation, an appropriate event listener is invoked. The event listeners validate the input itself or do it with the help of appropriate validators. Some inputs are simple to validate such as making sure whether a given field is a valid integer, real or text. However, making sure whether a given number is a valid road ID or making sure that adding or removing a road to an existing route will not make the route disconnected is a complex task. Special validators perform these complex validations. If a given user input is incorrect, an exception is thrown; otherwise, appropriate update manager is invoked by event listeners. The update manager then stores changes in OpenDRIVE and xProject files accordingly. There are four types of specialized validators, namely, Geometry Validator, Road Field Validator, Lane Validator, and Junction Validator as presented in Figure 20.

Geometry Validator is responsible for validating geometry of a road, which includes start point, length, curvature, and width. Drawing Area Popup Listener and Geometry Panel Listener invoke it because both of these listeners handle changes in road geometry. Junctions Listener invoke Junction Validator. The validator makes sure that definition of the junction is correct. Lane Panel Listener invokes Lane Validator. Lane Validator is responsible to check whether a lane added or modified by a user has valid physical properties such as maximum speed limit and lane width. Road Fields Panel Listener invokes Road Field Validator. This validator is invoked when a road is attached to a junction. The validators check whether such junction exists in the traffic scenario or not.

3.4.4 | Updaters

Upon validation of a CRUD operation, the listener invokes an appropriate updater. The Updater store changes in OpenDRIVE and xProject file. There are five updaters, namely, Road Fields Updater, Lanes Updater, Geometry Updater, Link Updater, and Junctions Updater. Road Fields Panel Listener invokes Road Fields Updater. The updater stores road definition in OpenDRIVE and xProject files. Geometry Panel Listener invokes Geometry Updater. The

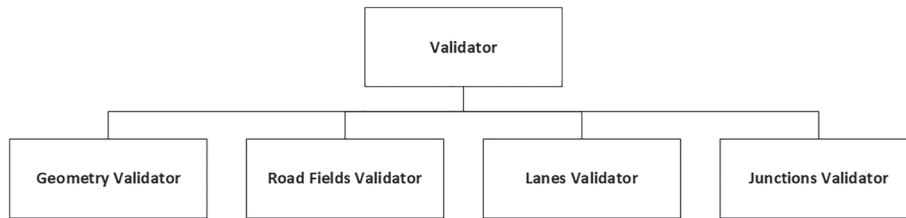


FIGURE 20 Architecture of validators

updater stores geometry of a road in OpenDRIVE file. Link Updater updates links between lanes of roads. Junctions Listener invokes Junctions Updater. Once junction information is validated, it is stored in OpenDRIVE by Junctions Updater. Architecture of updaters is presented in Figure 21.

3.4.5 | Exceptions

If a user requests an invalid CRUD operation, it may make an OpenDRIVE or xProject file semantically incorrect. Hence, these CRUD operations are validated before updating changes in OpenDRIVE and xProject files. If the CRUD operation is invalid, then an exception is thrown. Upon receiving an exception, the listener notifies user so that the user may amend the CRUD operation. There are three types of exceptions. Invalid Input Exception, Lane Exception, and Not Found Exception. Whenever user provides an invalid data type in a field, an Invalid Input Exception is thrown. For instance, if an input should be an Integer and user has provided a sequence of alphabets, this exception is thrown, and consequently, user is intimated. There are few user interactions where user must provide a valid lane number. If user does not comply to provide a valid lane number, then Lane Exception is thrown. There are interfaces that require a user to provide a valid road, junction, or signal by mentioning their ID. If there is no such component with user provided ID, then a Not Found Exception is thrown. Architecture of exceptions is presented in Figure 22.

3.5 | Training and demonstration resources

MovSim provides 57 different traffic scenarios so that domain experts may learn use of MovSim by experimenting with them. OpenDRIVE and xProject file of each scenario is provided with MovSim source code. MovSim provides traffic scenarios to demonstrate on ramp, start and stop phenomena using all reference models. These traffic scenarios have also been discussed at length in a book.⁴⁴ It is worth noticing that the authors of the book are also authors of MovSim. MovSim provides traffic scenario in which basic building blocks of a road network are demonstrated. MovSim also provides traffic scenarios that are designed to demonstrate different features of the simulator. However, to use MovSim,

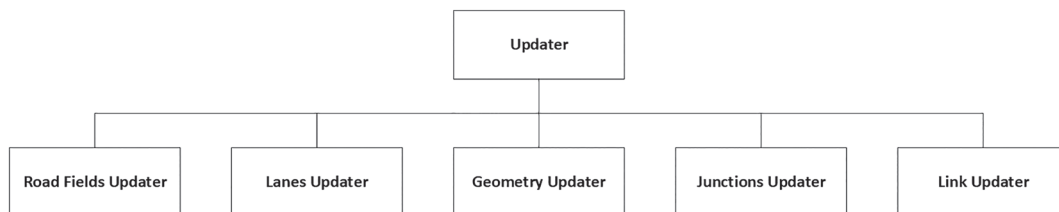


FIGURE 21 Architecture of updaters

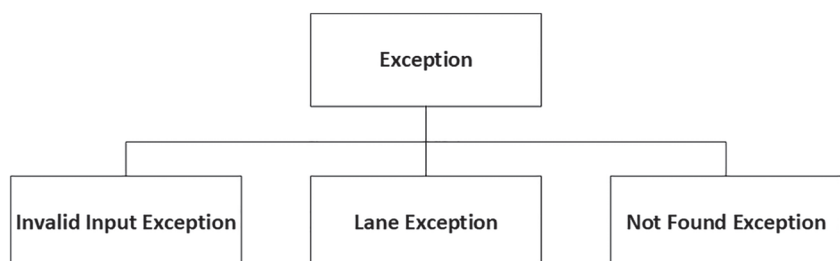


FIGURE 22 Architecture of exceptions

one should understand OpenDRIVE and xProject specifications. One should also be comfortable with XML technology. ITE bridges this gap by providing a readily available, easy-to-use, drag and drop-based GUI.

To demonstrate functionality of ITE, a traffic scenario, namely, clover leaf, has been developed using ITE. Clover leaf scenario has been selected for demonstration because its road network is large and complex. It helps in demonstrating ITE comprehensively. After making the scenario with ITE, OpenDRIVE and xProject files of the scenario have been compared with the one provided with MovSim. The XML files produces by the ITE are semantically equivalent to the one provided by the MovSim. Complete process of developing the cloverleaf scenario using ITE has been recorded. It is available online as training and demonstration resources of ITE on following URL in the form of videos (<https://goo.gl/DxbtGQ>).

3.6 | Software distribution

The ITE is free and open-source. Its code-base could be acquired from following URL using git client and could be built using Maven build management system (<http://github.com/AIMRL/TDE>). To acquire and build executable of ITE or any open-source software, one should have practical understanding of build management systems such as Maven, Ant, or Gradle.⁴⁵⁻⁴⁷ Usually, domain experts do not have these skills. To address this issue, along with code-base of ITE, readily usable executable for Windows platform is also provided. So that domain experts may download executable of ITE and design their own traffic scenarios using easy, drag and drop-based GUI. Through this, the ITE helps domain experts in conducting simulation-based experiments without being exposed to learning curve of Java, Maven, XML, OpenDRIVE, and xProject specifications. For now, the readily usable executable is provided for Windows platform only; however, to reach a wider user-base, readily usable executables for Mac OS and Linux platform would also be provided in the near future.

4 | CONCLUSION

Road transportation is very challenging due to the number of participants. Effective design and optimization of road transportation are important due to its role in sustainable development of society. To effectively study the transportation system, computational modeling and simulation could be effective as it is widely used to study complex systems. Simulation-based experiments are heavily employed in transportation research to comprehend its dynamics. MovSim is a free and open-source multimodal microscopic traffic simulator, which could be used to conduct studies to evaluate innovative designs and policies. Currently, to simulate a traffic scenario using MovSim, two XML files are required as input, one OpenDRIVE and the other xProject specification file. But domain experts are usually not comfortable with XML technology. Another issue is that building executable of MovSim from its source code is a very tedious job. To address these issues, a free and open-source tool, namely, ITE, have been developed. It assists domain experts, by providing a readily useable executable of ITE for Windows platform. The tool is also supplemented with a comprehensive video tutorial. The tool would help domain experts allocating more of their time on transportation science and engineering instead of issues related to software build management and understanding of OpenDRIVE and xProject specifications. ITE's GUI would help domain experts in developing complex traffic simulation scenarios efficiently using a variety of reference microscopic vehicular traffic models.

In the future, it is intended to develop an open-source module to plot results of simulation using common visualization techniques used in transportation related studies. It is also a useful extension that any model given in MovSim could be tuned against empirical data using different machine learning algorithms. Another extension is to develop an open-source module to visualize simulation in 3D. The module would provide technological basis to design an experimental setup to expose a simulated environment to a human subject using virtual reality headset and input gear that is used to play car racing games. This extension would help domain experts to conduct research on driving behavior of humans in integrated manner. It is also intended to incorporate Open Street Map into ITE so that simulation-based experiments could be performed on real road infrastructures (road maps).

Currently, most of the GUI components takes input as numerical values. Therefore, to effectively use the current GUI, fair amount of comprehension of OpenDRIVE and xProject formats is required. In the future, the GUI would be updated with more intuitive gadgets so that appropriate option could be represented in more semantic manner, and user may easily perform selection among the available options.

To gauge ITE's acceptability among the domain experts, it is planned that in the future, System Usability Scale (SUS) would be used. It is an elegant and simple method to empirically gauge whether a system is usable for intended user-base.

In the future, comprehensive tutorials would be made for domain experts to learn both MovSim and ITE. After training phase, domain experts would be given transportation science and engineering-related working tasks, which the domain experts would complete using both tools. On completion of this tasks, domain experts would be asked to assess both tools using SUS. Through this, an empirical evidence would be collected about the overall effectiveness of the ITE.

ORCID

Mian Muhammad Mubasher  <https://orcid.org/0000-0002-8450-0621>

Syed Waqar Jaffry  <https://orcid.org/0000-0003-4724-1752>

Muhammad Murtaza Yousaf  <http://orcid.org/0000-0001-9578-8811>

Imran Sarwar Bajwa  <https://orcid.org/0000-0002-5161-6441>

Shahzad Sarwar  <http://orcid.org/0000-0003-3074-9162>

Laeq Aslam  <http://orcid.org/0000-0002-1849-4606>

REFERENCES

- Henderson V. Urbanization in developing countries. *World Bank Res Obs*. May 2002;17(1):89-112.
- Treiber M, Kesting A, Thiemann C. *Traffic Flow Dynamics: Data, Models and Simulation*. Berlin Heidelberg: Springer; 2012. <https://www.springer.com/gp/book/9783642324598>
- Richardson GP. System dynamics: simulation for policy analysis from a feedback perspective. In: Fishwick PA, Luker PA, eds. *Qualitative Simulation Modeling and Analysis*. New York: Springer; 1991:144-169. https://link.springer.com/chapter/10.1007/978-1-4613-9072-5_7
- Harder BN. Use of simulation in teaching and learning in health sciences: a systematic review. *J Nurs Educ*. Jan. 2010;49(1):23-28.
- Mubasher M, Jaffry W. Systematic literature review of vehicular traffic flow simulators, in 2015 International Conference on Open Source Software Computing (OSSCOM), 2015, pp. 1–6.
- Ejercito PM, Nebrija KGE, Fera RP, Lara-Figueroa LL. Traffic simulation software review, in 2017 8th International Conference on Information, Intelligence, Systems Applications (IISA), 2017, pp. 1–4.
- Treiber M, Kesting A. An open-source microscopic traffic simulator. *IEEE Intelligent Transportation Systems Magazine*. 2010;2(3):6-13, Fall.
- Omahen A, Kukuljan T, Vida V, Peter K, Papa G. Upgrade of the MovSim for easy traffic network modification. presented at the SIMUL 2015: The Seventh International Conference on Advances in System Simulation, Barcelona, Spain, 2015.
- Bonaventura M, Wainer GA, Castro R. Advanced IDE for modeling and simulation of discrete event systems, in *Proceedings of the 2010 Spring Simulation Multiconference*, San Diego, CA, USA, 2010, pp. 125:1–125:8.
- Takahashi M, Nakanishi T, Miyoshi I, Fujikura T. An evaluation of the road traffic system simulator PIMTRACS by PIM. *Mathematics and Computers in Simulation*. May 2002;59(1-3):45-56.
- Barcelo J, Ferrer J, Montero L. AIMSUN: advanced interactive microscopic simulator for urban networks. User's Manual, Departament d'Estadística i Investigació Operativa, UPC, 1997.
- Liao C-F, Morris T, Donath M. Development of internet-based traffic simulation framework for transportation education and training. *Transportation Research Record: Journal of the Transportation Research Board*. Jan. 2006;1956(1):184-192.
- Liao C-F, Morris T, Donath M. Web-based traffic simulation framework for ITS education and training. presented at the Intelligent Transportation Society of America - 12th World Congress on Intelligent Transport Systems 2005, 2009, vol. 7, pp. 4260–4269.
- Liao C-F, Levinson DM. ROAD: interactive geometric design tool for transportation education and training. *Journal of Professional Issues in Engineering Education and Practice*. 2013;139(2):116-122.
- Debiasi A, Prandi F, Conti G, De Amicis R, Stojanović R. Visual analytics tool for urban traffic simulation, in *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques*, ICST, Brussels, Belgium, Belgium, 2013, pp. 51–56.
- Dupuis M. OpenDRIVE format specification revision 1.4. VIRES Simulationstechnologie GmbH, 1.4, Nov. 2015.
- OpenDRIVE (specification). *Wikipedia*. 03-Mar-2017.
- Hiblot N, Gruyer D, Barreiro J-S, Monnier B. Pro-SIVIC and ROADS. A software suite for sensors simulation and virtual prototyping of ADAS, in *Proceedings of DSC*, 2010, pp. 277–288.
- Haubrich T, Seele S, Herpers R, Becker P. Integration of road network logics into virtual environments, in 2014 IEEE Virtual Reality (VR), 2014, pp. 79–80.
- Tejkl K, Ressel W. Virtual road models from dynamic measurements, in *Proceedings of the International Conference on Road and Rail Infrastructure CETRA*, 2012.
- Haubrich T, Seele S, Herpers R, Müller ME, Becker P. A semantic road network model for traffic simulations in virtual environments: generation and integration, in 2014 IEEE 7th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS), 2014, pp. 43–50.

22. Richter A, Friedl H, Scholz M. "Beyond OSM—alternative data sources and approaches enhancing generation of road networks for traffic and driving simulations," in *Proceedings of the SUMO2016. Ther Ber.* 2016;30:23-31.
23. Treiber M, Hennecke A, Helbing D. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E.* Aug. 2000;62(2):1805-1824.
24. Bando M, Hasebe K, Nakanishi K, Nakayama A. Analysis of optimal velocity model with explicit delay. *Phys. Rev. E.* Nov. 1998;58(5):5429-5435.
25. Jiang R, Wu Q, Zhu Z. Full velocity difference model for a car-following theory. *Phys. Rev. E.* Jun. 2001;64(1):017101.
26. Gipps PG. A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological.* Apr. 1981;15(2):105-111.
27. Krauss S, Wagner P, Gawron C. Metastable states in a microscopic model of traffic flow. *Phys Rev E.* May 1997;55(5):5597-5602.
28. Nagel K, Schreckenberg M. A cellular automaton model for freeway traffic. *Journal de Physique I.* Dec. 1992;2(12):2221-2229.
29. Kerner BS, Klenov SL, Wolf DE. Cellular automata approach to three-phase traffic theory. *J Phys a: Math Gen.* Nov. 2002;35(47):9971-10013.
30. Kesting A, Treiber M, Helbing D. General lane-changing model MOBIL for car-following models. *Transportation Research Record.* Jan. 2007;1999(1):86-94.
31. Riley D, Nellen G, Barrera R, Quevedo J. Crowdsourcing traffic simulation to improve signal timing. *J Comput Sci Coll.* May 2014;29(5):112-118.
32. Treiber M, Germ R, Kesting A. From drivers to athletes: modeling and simulating cross-country skiing marathons. In: *Traffic and Granular Flow'13.* Cham: Springer; 2015:243-249.
33. van Willigen WH, Haasdijk E, Kester LJHM. Evolving intelligent vehicle control using multi-objective NEAT, in 2013 IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems (CIVTS), 2013, pp. 9–15.
34. van Willigen W, Haasdijk E, Kester L. Fast, comfortable or economical: evolving platooning strategies with many objectives, in 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), 2013, pp. 1448–1455.
35. Mubasher M, Jaffry W. Incorporation of the driver's personality profile in an agent model. *PROMET - Traffic&Transportation.* Dec. 2015;27(6):505-514.
36. Guériaux M, Billot R, Hassas S, Armetta F, Faouzi NEE. An extension of MovSim for multi-agent cooperative vehicles modeling, in 2014 International Conference on Connected Vehicles and Expo (ICCVE), 2014, pp. 859–860.
37. Jeyabalan J, Soundarajan SS, Krishnan P. A hybrid model for sparsely populated VANETs, in Proceedings of IEEE International Conference on Computer Communication and Systems ICCCS14, 2014, pp. 030–032.
38. Guériaux M, Billot R, Faouzi N-EE, Hassas S, Armetta F. Multi-agent dynamic coupling for cooperative vehicles modeling, in *The Twenty-Ninth Conference on Artificial Intelligence AAAI'2015 - (DEMO Track)*, Austin, United States, 2015, pp. 4276–4277.
39. Guériaux M, Billot R, El Faouzi N-E, Monteil J, Armetta F, Hassas S. How to assess the benefits of connected vehicles? A simulation framework for the design of cooperative traffic management strategies. *Transportation Research Part C: Emerging Technologies.* Jun. 2016;67:266-279.
40. Yan X, Gu F, Hu X, Engstrom C. Dynamic data driven event reconstruction for traffic simulation using sequential Monte Carlo methods, in 2013 Winter Simulations Conference (WSC), 2013, pp. 2042–2053.
41. Xue H. Data assimilation based on sequential Monte Carlo methods for dynamic data driven simulation, Computer Science Dissertations, Aug. 2014.
42. Wu P, Xue H, Hu X. Particle filter based traffic data assimilation with sensor informed proposal distribution, in *Proceedings of the 48th Annual Simulation Symposium*, San Diego, CA, USA, 2015, pp. 173–180.
43. Feng X, Yan X, Hu X. Dynamic data driven particle filter for agent-based traffic state estimation. *Cloud Computing and Security.* 2015;9483:321-331. https://link.springer.com/chapter/10.1007/978-3-319-27051-7_27
44. Treiber M, Kesting A. *Traffic Flow Dynamics: Data, Models and Simulation.* Springer; 2012.
45. Company S. *Maven: The Definitive Guide.* 1st ed. Beijing: O'Reilly Media; 2008.
46. Tilly Jesse EMB, Ant: *The Definitive Guide.* Sebastopol, California, United States: O'Reilly Media; 2005. <http://shop.oreilly.com/product/9780596001841.do>
47. Dockter Matthew TB, McCullough H. *Building and Testing with Gradle.* Sebastopol, California, United States: O'Reilly Media; 2011. <https://www.oreilly.com/library/view/building-and-testing/9781449306816/>

How to cite this article: Mubasher MM, Jaffry SW, Yousaf MM, Bajwa IS, Sarwar S, Aslam L. A smart integrated environment for vehicular traffic simulation. *Int J Commun Syst.* 2019;e4029. <https://doi.org/10.1002/dac.4029>