Microscopic Driver-Centric Simulator: Linking Unity3D and SUMO

Carlos Biurrun-Quel[1], Luis Serrano-Arriezu[1] and Cristina Olaverri-Monreal[2*]

[1] Universidad Pública de Navarra, Dept. Electrical and Electronic Engineering
E-31006 Pamplona, Navarra, Spain
biurrun.79094@e.unavarra.es, lserrano@unavarra.es

[2] University of Applied Sciences Technikum Wien, Department of Information Engineering and Security, Vienna, Austria
olaverri@technikum-wien.at

Abstract. Many different tools have been developed for traffic simulation. These tools allow the representation of complex, realistic traffic situations that can be useful in evaluating a specific traffic situation or testing new technological applications and their influence on the driver in real traffic scenarios. However, nowadays not many examples of driver-centric driving platforms exist in which the mobility behavior of other vehicles is based on traffic models. In this work, the elaboration of a 3D Driver-Centric Simulator capable of using real-world road networks together with realistic traffic models is presented. Implementation of TraCI protocol allows communication between Unity3D and the microscopic traffic simulator SUMO to provide driving conditions. An evaluation of the simulator's performance is presented and future lines of work are defined.

Keywords: Driver-Centric, SUMO, Unity3D, TraCI, Microscopic Simulation

1 Introduction

Vehicular Ad-Hoc Networks (VANETs) allow vehicles to communicate among each other, as well as communication between vehicles and roadside units. Together with VANETs, several applications are being developed, such as In-Vehicle Information Systems (IVIS) and Advanced Driver Assistance Systems (ADAS). These applications present many different potential benefits, from improving driver safety and driving performance, to providing driver comfort and entertainment, to enhancing the use of transport networks, thereby increasing their efficiency.

These systems may also be a potential source of distraction for the driver, which is why their testing and assessment stands as an important phase during the development process. However, deployment and testing of these applications in a real environment

[*] Corresponding author

This is a pre-copyedited version of a contribution published in: Rocha Á., Correia A., Adeli H., Reis L., Costanzo S. (eds) Recent Advances in Information Systems and Technologies, WorldCIST 2017. Advances in Intelligent Systems and Computing, vol 569 published by Springer, Cham. The definitive authenticated version is available online via https://doi.org/10.1007/978-3-319-56535-4_83

entail some potential risks, such as endangering not only the test driver, but also surrounding individuals, while also requiring a considerable investment in terms of money and time. On the other hand, simulation presents a reliable, low-cost and low-risk solution which additionally provides quick feedback and a high level of flexibility. Thus, a driver-centric simulator which generates NPC[1] traffic based on realistic traffic models stands as a powerful testing tool for the technologies mentioned above.

Traffic simulation tools can generate scenarios where vehicles are modelled by complex mathematical methods. Thus, employing a reliable simulation tool to generate traffic for a driving scenario stands as one of the fundamental steps toward generating driver-centric simulators.

In this work the development of a driver-centric traffic simulator is presented. This application is built over a 3D graphic engine which generates the driving scenario at the same time it communicates dynamically with a traffic simulator in order to mimic real traffic conditions. We contribute to the state of the art locally implementing the TraaS (TraCI as a Service) library and performing the interaction between the user controlled vehicle and SUMO.

The following sections are organized as follows. Section 2 provides some related literature and basic concepts of driving simulation; Section 3 presents the implementation of the tool, including the generation of the roads and vehicles and the implementation of the TraCI protocol. An evaluation of the tool's performance is found in Section 4. Conclusions and some future lines of work are defined in Section 5.

2 Related Work

Simulation platforms are remarkable and very effective tools for analyzing different scenarios which cannot be studied by other means or whose cost are not be affordable in a given project. They base their calculations on one or more mathematical approaches and algorithms called simulation models. This section presents a brief review of traffic simulators.

2.1 Microscopic Traffic Simulators

In the past decades, several microscopic simulation tools have been developed. Some of them are the result of research projects and are currently available as open-source tools, while others are exclusively commercial and a license should be paid in order to use them. AIMSUN [1] is a hybrid traffic modelling software which allows simultaneous application of multimodal (capable of simulating different transport modes) microscopic and mesoscopic analysis to large networks, enabling the modelling of each area with an adjustable level of detail. It has a detailed Graphical User Interface (GUI) which also allows creation of a 3D representation of the simulation. VISSIM [2] is one of the microscopic simulators that are leading the market nowadays, with more than 7,000 licenses worldwide. It allows one to model multimodal traffic flows

---

1 No-Player-Controlled

in urban areas, as well as on motorways and rails. It also includes a complete GUI with a toggle button enabling 3D perspective for the simulation process. PARAMICS [3] is another commercial, microscopic, multimodal simulation tool. It allows traffic and pedestrian simulation, constructs large networks and offers detailed reports of Measures of Effectiveness. Similarly to the aforementioned simulator, PARAMICS also includes the possibility to enable a 3D representation of the simulation in the GUI. SUMO [4] (Simulation of Urban Mobility) is a microscopic, multimodal, open source simulator that has been under development for more than 15 years, in large part by the Institute of Transportation Research (IV) at the German Aerospace Centre (DLR), although there continue to be additional contributors. One remarkable feature is the implementation of the TraCI protocol [5], which allows data retrieval and remote control simulation through a Transmission Control Protocol (TCP) connection from another application. Importation of real maps from sources like Open Street Map (OSM) [6] is also available.

2.2 Linked Simulation Platforms

Previous works that couple a traffic simulator together with other tools have been carried out. One of the most common approaches in the field of Intelligent Transport Systems (ITS) is the recent development of so-called VANET simulators. These tools are made by coupling one traffic simulator together with a network simulator, where the different nodes of the network correspond to vehicles communicating among themselves and with Road Side Units (RSUs). Authors in [7] manage to get a traffic simulator, SUMO and a network simulator, NS2 [8], to work together thanks to communication through TraCI protocol. Vehicles are translated into nodes which communicate among themselves, forming a VANET inside the network simulator. The interaction between nodes and the TraCI interface allow modification of the mobility behavior of a single vehicle inside SUMO. A similar approach, Veins, is presented in [9], where bidirectional coupling between network simulator OMNeT++ [10] and SUMO is achieved by establishing communication between both tools through a TraCI interface. Although none of these approaches involve a driver-centric scenario, they do provide reliable examples of the implementation of the TraCI protocol to modify the mobility behavior of the vehicles in SUMO from external applications. In recent work from the Royal Institute of Technology of Stockholm [11] a way of communicating between the traffic simulator SUMO and the graphic 3D engine Unity3D [12] was developed in order to satisfy the need for a realistic visualization of traffic in urban environments. For this purpose, the TraCI protocol was used in order to link SUMO and Unity3D. Nonetheless, no approach with a driver-centric perspective was carried out. Therefore, driver behavior cannot influence the other vehicles. The goal of coupling a traffic simulator together with a 3D engine is achieved in [13], and is developed using OpenSceneGraph [14] as the 3D Engine, DIVERT [15] as the traffic simulator, and NS-3 as a network simulator. In this work, authors used this tool in order to implement and evaluate Virtual Traffic Lights [16] and See-Through System [17] applications inside the vehicle.

2.3 Driver-Centric Simulators

IC-DEEP (In-Car Ergonomics Evaluation Platform) is an implementation approach from a driver-centric perspective in form of a Serious Game (SG). It was developed to autonomously assess, under low-time and low-cost conditions, the factors that can jeopardize the driving performance when manipulating or receiving information from in-vehicle information systems (IVISs) [18], [19]. An improved version of the framework built on modular components was implemented to efficiently configure new experiments and to make it easier to upgrade and update the experiment scenario and overall conditions [20].

A further driver-centric simulator was presented in [21]. In this work, the SUMO simulator is used to generate traces of the vehicle movements, which are used by Unity3D to generate the model traffic. However, this implies that no direct communication between SUMO and Unity3D is being done, since traces are processed offline.

3 Implementation

To implement our driver-centric simulator, we distinguish two main parts: a microscopic traffic simulation tool, which can generate traffic flows based on well-known, reliable, complex traffic models; and a 3D graphic engine that allows the construction of the 3D scenario. SUMO was chosen as the source traffic simulation tool for the development of our driver-centric simulator due to its open-source availability, the feedback from several works that have successfully employed this tool, and the implementation of the TraCI protocol, which enables the data retrieval regarding number of vehicles, their individual speed and position, and the shape of the roads. Data from this tool is extracted through TraCI Protocol and processed by a 3D graphic engine. After considering other tools (e.g. OpenSceneGraph) Unity 3D was selected as a suitable tool for developing our driving simulator due to the availability of a cost-free version, its powerful engine, the enormous amount of documentation and online support, as well as a successful track record in other projects. The following section presents first an elaboration of the test scenario in SUMO, importing real urban maps and generating traffic. Then, the connection between SUMO and the 3D scenario itself is addressed.

3.1 Generation of the Scenario in SUMO

The traffic network in SUMO[2] is defined by the SUMO network file. This file has an extension .net.xml specific to SUMO. An X-Y coordinate system is used to reference the position of all the elements of the network. The origin of this system is the (0,0) point, and the positive Y axis is oriented north (the other 3 cardinal points are oriented

[2] The latest version of SUMO is 0.26 but because this version was launched during the elaboration of this work, it was not used in the current project. Available at the start of the project, version 0.25 was used.

subsequently). This must be taken into account when manually defining a network, analysing an existing network and translating this network to Unity3D.

Generating the Network.

SUMO offers a variety of ways for generating the network. The simplest, but also most tedious, method is to manually create the XML files for structuring the network. However, this method is not scalable and very time consuming. In addition, since the proposed driver-centric simulator aims to evaluate realistic situations, networks representing real existing roads are required. To fulfil these requirements, data from real, existing roads must be used. One useful source of this type of data is OSM. A specific area can be selected and exported to XML format. The selected area contained the surroundings of the UAS Technikum Wien in the city of Vienna.

The NETCONVERT tool, included in the SUMO package, was used to generate the corresponding SUMO file. The generated network includes not only roads but also railways and sidewalks, and some incomplete roads might be additionally generated. The console version of the NETCONVERT tool displays a series of warnings and errors that explain these problems. For this reason, it was determined that the network required a more precise editing. This could be achieved by using the NETEDIT tool, included as well in the SUMO package. This tool allows the edition of the network through a GUI in a very simple and user-friendly way. The result after the edition process can be found in Fig. 1. Extra railways, nodes and edges were removed, and traffic signal programs were rescheduled, since the original did not match in all the intersections. In addition, an extra edge was created as a reference for the starting point for the controlled vehicle in the driver-centric simulator.

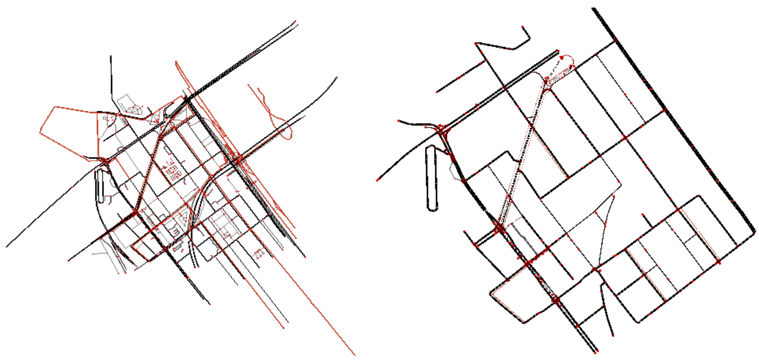

Fig. 1. Road network before (left) and after edition (right)

Generating the NPC Traffic.

Once the network was created, the generation of some automatically controlled vehicles was required in order to provide some traffic conditions in the simulator. Since traffic generation was not considered a key factor for a first test of the driver-centric simulation tool, the script randomTrips.py and the DUAROUTER tool, both included in the SUMO package, were used in order to create some random trips of at least 500 meters across the network.

3.2 Communication between Unity3D and SUMO – TraCI Protocol

TraCI, or Traffic Control Interface, is the communication protocol that allows control and modification of a simulation process concurrent with SUMO. Furthermore, it also allows the retrieval of any of the elements of the network that are being used in the simulation, such as vehicles, edges, junctions, or traffic lights. In addition, each of these objects has a series of variables which can be retrieved or modified online. Communication between SUMO and a TraCI client follows a request-reply pattern over a TPC/IP connection.

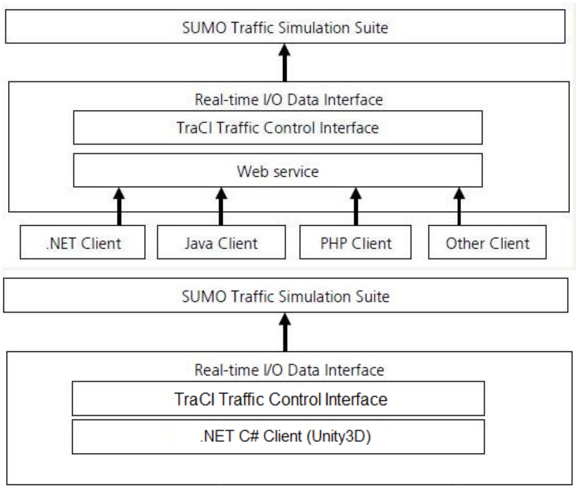

Fig. 2. TraaS Web Service implementation (top). TraaS implementation in Unity3D (bottom)

First, SUMO must be run as a TraCI Server, listening to a specific port. Once SUMO accepts the connection, the simulation control duties are transferred to the client. From this moment, the client can determine when to perform the simulation steps and decide whether to apply changes to vehicle variables or traffic signal logics, reassign the vehicle routing, or retrieve data regarding these elements.

In order to implement this protocol, an existing library, TraaS (TraCI as a Service) [22], was chosen. It is an open source web service which allows remote control of SUMO from any programming language. However, as explained in [22], making use of a web server would introduce a considerable amount of delay in the proposed driver-centric simulator, and some synchronization problems might occur. Therefore, the implementation of this library was made local. Fig. 2 shows the comparison diagrams between the original web service implementation and the developed one.

3.3 Generation of the Scenario with Unity3D

Two main parts are distinguished in the implementation of the driver-centric simulator in Unity3D: initialization and update. The initialization process retrieves information regarding the roads that belong to the network and the generation of the user-controlled vehicle, the key object for the driver-centric simulator. One object of the class "SumoTraciConnection" handles the execution and communication with SUMO. Therefore, it launches a new instance of SUMO as a TraCI server, connects to it as a client and controls the simulation execution as well as the data exchange. The update process controls the simulation, periodically retrieving the position of every NPC vehicle, updating their position in the 3D scenario and, if needed, inserting new vehicles or deleting them when they finish their trips.

Unity3D employs the method "FixedUpdate()" to change the scene to the following frame ensuring a constant number of frames-per-second (FPS). This implies that the gap between simulation steps remain constant, collaborating to achieve synchronization. However, it was observed that performing a request of the vehicles in the network on every frame was quite computationally demanding, and at the same time it resulted in a very unrealistic movement. Since the utilized simulation step in SUMO corresponded to 100 milliseconds, the vehicles seemed to teleport from one location to another on every frame. In order to reduce the number of requests, these where set every 10 frames. Then, in order to fix the difference between simulation steps, position of the vehicles was estimated every two frames using the available speed of each. In the end, a smooth, realistic movement of the vehicles was achieved. The update process also involves the control of the user driving process. On every frame, inputs are verified, their equivalent acceleration/braking/turning power is computed, and the new speed and position are calculated.

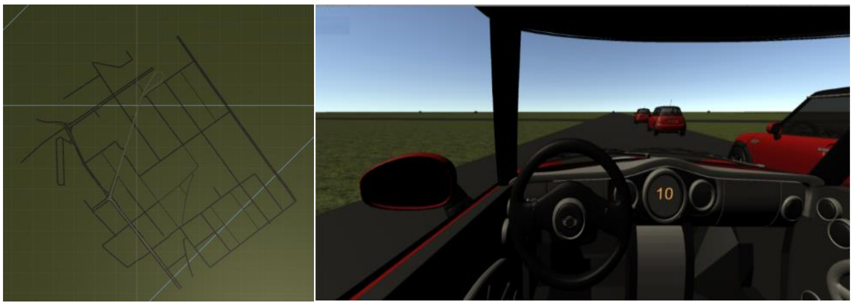

Fig. 3. Bird's-eye view of the scenario in Unity3D (left). Driver-centric perspective of the scenario (right).

An aerial view of the generated road network in Unity3D is provided in Fig. 3 left. The driver-centric perspective is presented in Fig. 3 right. A complete view of the left-side mirror, windshield and steering wheel is provided. A display also presents the speed in km/h. The model used for the controlled car is the same as NPC vehicles. Road width and length had to be assigned proportionally to the vehicle's size in order to keep commensurate with the network size.

4 Evaluation of Performance

In this section, some comments regarding the simulator performance are provided. These comments concern the three main parts of the simulation: generation of the roads, generation of the vehicles and driving performance. Since OSM data, an open-source and not error-free, resource has been used, the generated networks might be not reliable. Knowledge of the extracted areas might help determine the degree of reliability of the obtained network. Furthermore, the use of an automatic tool in order to translate the OSM data into a SUMO network file was observed to introduce some minor mistakes, which had to be fixed using the NETEDIT tool. When representing these mistakes on the 3D scenario, wrong representations such as those seen in Fig. 4 might occur.

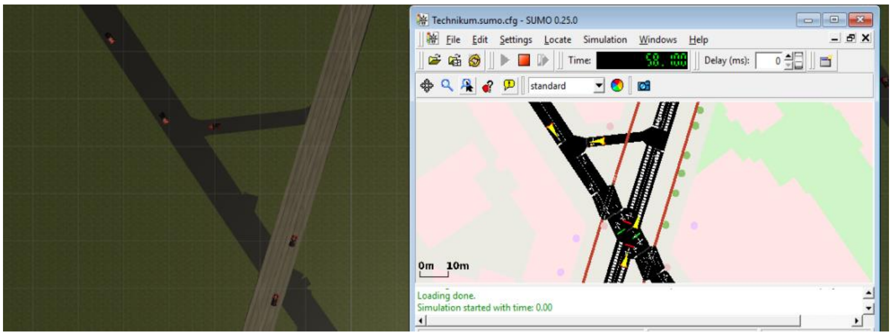

Fig. 4. Example of a distorted intersection in the SUMO network represented in Unity3D

These errors were found especially in areas with many intersections nearby and can be fixed by editing the network in SUMO. As for the vehicle generation, an automatic tool for random trip generation was used in order to provide simple traffic flows throughout the scenario. For a more professional approach with a specific case study, a higher level of detail is required. Finally, the performance of the driving task in terms of delay was good except for the cases in which the amount of NPC vehicles contained in the scenario was approximately over 50. In those cases, the driving performance diminished highly due to lack of computational power, causing the simulation to lag and making it unfeasible to drive.

5 Conclusion and Future Work

In this work, the development of a driver-centric simulator where traffic is generated by following microscopic models has been presented. This simulator is based on the connection between the microscopic traffic simulator SUMO and the 3D graphic engine Unity3D though TraCI protocol. Randomly generated trips have been created in order to perform a first test of the simulator. While this approach is yet embryonic, the basic structure of the driver-centric tool makes it possible to develop case studies for testing new ITS applications. Further modifications to this tool can be made in order to improve realism and complexity of traffic generation as well as tool's performance. Future lines of work will address the inclusion of buildings, models for different vehicle types and a mechanism that only renders the vehicles located within a certain radius from the driver, in order to increase computational efficiency as well as the configuration of the interaction between NPC vehicles and the user- controlled vehicle. In addition, the TraCI protocol allows the retrieval of information from various types of objects, such as traffic lights programmed in SUMO, which can also be implemented in the 3D scenario. Additional computational power will make it possible to increase the number of NPC vehicles contained in the scenario.

Acknowledgments. This work was supported by the "KiTSmart Project – City of Vienna Competence Team for Intelligent Technologies in Smart Cities", project number 18-07 funded by national funds through the MA 23, Urban Administration for Economy, Work and Statistics, Vienna, Austria and the EU Erasmus + framework program.

6 References

1. "Top Features in Aimsun" 2015. [Online]. Available: https://www.aimsun.com/aimsun/top-features.
2. J. Barcelo, M. Fellendorf, and P. Vortisch, Fundamentals of Traffic Simulation, vol. 145. 2010.
3. Quadstone Paramics | "Traffic and Pedestrian Simulation, Analysis and Design Software," 2016. [Online]. Available: http://www.paramics-online.com/.
4. D. Krajzewicz, G. Hertkorn, "SUMO (Simulation of Urban MObility) An open-source traffic simulation," 2002.
5. A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "TraCI: An Interface for Coupling Road Traffic and Network Simulators AxelWegener1," Proc. 11th Commun. Netw. Simul. Symp. - CNS '08, pp. 155–163, 2008.
6. "OSM - Open Street Map," 2016. [Online]. Available: https://www.openstreetmap.org
7. M. Piórkowski, M. Raya, a. L. Lugo, P. Papadimitratos, M. Grossglauser, and J.-P. Hubaux, "TraNS: Realistic Joint Traffic and Network Simulator for VANETs," ACM SIGMOBILE Mob. Comput. Commun. Rev., vol. 12, no. 1, p. 31, 2008.
8. NS-2, Network Simulator, http://nsnam.sourceforge.net/wiki/index.php/Main_Page
9. C. Sommer, S. Member, and R. German, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis", vol. 10, no. 1, pp. 3–15, 2011.
10. Omnet, "Omnet++ discrete event simulator - home," https://omnetpp.org/, (accessed on: 25.07.2016).
11. J. Raghothama, M. Azhari, M. R. Carretero, and S. Meijer, "Architectures for distributed, interactive and integrated traffic simulations", 2015 Int. Conf. Model. Technol. Intell. Transp. Syst. MT-ITS 2015, no. June, pp. 387–394, 2015.
12. Unity3D, https://unity3d.com/
13. P. Gomes, C. Olaverri-Monreal, and M. Ferreira, "Driver-Centric VANET Simulation", Proceedings Nets4Cars 2011 International Workshop on Communication Technologies for

Vehicles. German Aerospace Center (DLR), Oberpfaffenhofen, Germany, pp. 143-154, 2011.
14. Openscenegraph, http://www.openscenegraph.org
15. R. Fernandes, P. D'Orey, M. Ferreira, "DIVERT for Realistic Simulation of Heterogeneous Vehicular Networks". In: 2nd IEEE International Workshop on Intelligent Vehicular Networks - InVeNET , 2011
16. C. Olaverri-Monreal, P. Gomes, M. Krüger Silvéria, M. Ferreira, "In-Vehicle Virtual Traffic Lights: a Graphical User Interface", Proceedings 7th Iberian Conference on Information Systems and Technologies, CISTI'2012, Madrid, Spain, 2012, pp. 1 - 6.
17. C. Olaverri-Monreal, P. Gomes, R. Fernandes, F. Vieira, and M. Ferreira, "The See-Through System: A VANET-enabled assistant for overtaking maneuvers," in 2010 IEEE Intelligent Vehicles Symposium, 2010, pp. 123–128.
18. J. Gonçalves, R. J. F. Rossetti, and C. Olaverri-Monreal, "IC-DEEP: A serious games based application to assess the ergonomics of in-vehicle information systems," Proceedings IEEE Conference on Intelligent Transportation Systems, ITSC 2012, pp. 1809–1814.
19. J. S. V. Gonçalves, R. J. F. Rossetti, J. B. Jacob, J. Gonçalves, C. Olaverri-Monreal, A.B. Coelho, and R. B. Rodrigues, "Testing Advanced Driver Assistance Systems with a serious-game-based human factors analysis suite," Proceedings IEEE Intelligent Vehicles Symposium, 2014 pp. 13–18.
20. C. Olaverri-Monreal, J. Gonçalves, K. Bengler, "Studying the Driving Performance of Drivers with Children Aboard by Means of a Framework for Flexible Experiment Configuration", Proceedings IEEE Intelligent Vehicles Symposium Workshops, Dearborn, Michigan, USA. pp. 7–12, 2014
21. V. Charissis, S. Papanastasiou, W. Chan, and E. Peytchev, "Evolution of a full-windshield HUD designed for current VANET communication standards," IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC, pp. 1637–1643, 2013.
22. "TraaS. TraCI as a Service." [Online]. Available: http://traas.sourceforge.net/cms/.