

INTEGRATION AND SYNCHRONIZATION OF DIFFERENT-SCOPE AUTOMOTIVE SIMULATORS FOR AUTONOMOUS VEHICLES' TESTING

Diego Fuentes, Jesús Rubén Gastón (HI-iberia Ingeniería y Proyectos SL, Spain), **Raúl Correal** (Ixion Industry & Aerospace SL, Spain), **Jorge Villagr a** (Centre for Automation and Robotics, CSIC-UPM, Spain)

Keywords: Autonomous Vehicles, Traffic, Simulation, Synchronization

Abstract

Traffic simulation and self-driving cars have been a very active research field in the Intelligent Transportation Systems community in the last years. The complexity of automated systems need exhaustive testing around a huge range of driving situations. This validation is often prohibitive to be done only by experimental means, and therefore simulation becomes crucial. In addition, there is a growing number of business models around connected and automated driving, such as Dynamic Ridesharing or Intelligent Parking Management, for which the only way to assess their added value is through a proper combination of microscopic and nanoscopic simulation capabilities. This paper proposes a novel approach to integrate and synchronize simulators of different granularity in the scope of automated and connected driving. It focuses on the coupling of an open-source microscopic simulation framework (SUMO) with an ADAS-oriented nanoscopic simulation tool (SCANer). An interface between both environments has been designed to make them consistent in the local area the ADAS simulator is evolving. As a result, the ego and surrounding vehicles evolution are part of a highly frequently updated closed-loop system between both environments.

1. Introduction

Traffic simulation has been a very active research field in the Intelligent Transportation Systems community in the last 30 years, providing solutions ranging from road network designs to urban traffic lights distribution. In the automotive industry the use of simulation is well established in the development process of traditional driver assistance and active safety systems. More recently, simulation-based validation tools have become one of the cornerstones for the rapid deployment of autonomous driving systems. For the latter, multiple simulation technologies that had progressed independently need to be seamlessly integrated.

Each simulation tool tries to answer the question “What would happen if” in real world [1]. Following the answer to that question, road traffic simulations can generally be subdivided into 4 categories according to the level of detail [2] [3]: macroscopic, mesoscopic, microscopic and nanoscopic. While macroscopic flow models describe traffic as aggregate flows, microscopic simulators model the behaviour and interactions of each simulated entity individually with a reduced set of state variables. Mesoscopic models are medium-detailed models where traffic is usually represented by queues of vehicles. In nanoscopic models, a higher level of detail is achieved introducing vehicle dynamics, complex decision processes of the driver or a precise interaction with the vehicle surroundings.

It is well known that the complexity of automated systems need exhaustive testing around a huge range of driving situations. As real testing for a simple function may need over 109 km before its validation, it is prohibitive to do it only by experimental means, and therefore simulation becomes crucial. However, the complexity of urban scenarios cannot be reproduced

with a single type of simulator (macroscopic or microscopic ones cannot incorporate the behaviour of these functions, while nanoscopic tools cannot handle all the heterogeneity and uncertainty of complex driving scenarios, such complex roundabouts or intersections).

This is the main reason why there is an increasing interest in putting together already existing realistic nanoscopic simulators for Advanced Driving Assistance Systems (ADAS) with microscopic traffic simulators.

In addition to that, there is a growing number of business models around connected and automated driving, such as Dynamic Ridesharing or Intelligent Parking Management, for which the only way to assess their added value is through a proper combination of microscopic and ADAS nanoscopic simulators.

This paper proposes a novel approach to integrate and synchronize simulators of different granularity in the scope of automated and connected driving. It will in addition take advantage of an open simulation paradigm (SUMO), with constant updates and collaboratively developed plugins.

The results which are presented are the outcomes of the research activities performed under the framework of the JTI Artemis project "Embedded multi-core systems for mixed criticality applications in dynamic and changeable real-time environments (EMC2)".

2. Related Works

Different traffic simulators [4], communications simulators [5], automotive-oriented simulators [5] [6] [7], robotics-inspired simulators [9][10] and driver simulators [11] have been used so far as independent tools. As the increasing complexity of autonomous driving systems require interfacing between some of them, some preliminary works have recently been conducted. In this connection, the pioneer works [12][13] have arisen in the area of cooperative intelligent transportation systems, with the connection between well-known microscopic traffic simulator (SUMO [4], VISSIM [14]) and communication simulation tools NS-2 [5]. VsimRTI [15] enables also the preparation and execution of V2X-based traffic simulations. A different proposal is the general framework MOBYSIM [16], an integrated simulation platform that combines a number of tools including PreScan [7] and OPNET.STRAW [17].

In parallel, the advances in the generation of very realistic worlds and sensor models have allowed in the last decade a scope of different possibilities for ADAS validation. Indeed, the already existing vehicle dynamics simulation tools have been enhanced with these new 3D processing technologies. As a result, very representative driving scenarios, including the interactions between the ego-vehicle and its direct surroundings, can be realistically emulated. The main disadvantage of this type of tool (nanoscopic or sub-microscopic) is that it is limited to few situations and it is not easy to find real mutual interaction among all the road agents in the scene. In addition to that, when simulating large scale scenarios, the computational resources required can quickly become prohibitive.

To overcome these limitations, the best of traffic simulators (they get a very realistic interaction model for large-scale driving scenes, while having a reduced number of parameters, minimizing thus the effort of calibration) can be merged with the aforementioned advantages of nanoscopic simulation tools [18]. Some previous examples in this direction are PARCOURS [15], integrating 3D virtual reality driving simulation with SUMO, or the work conducted within the European project PREDRIVE C2X, where an integrated simulation framework for cooperative automated vehicles was designed. In this connection, the MiReCol [19] general software architecture was developed for the integration of simulation models and real world

data in a much wider context than mobility alone. Another relevant work is MDDSVsim [20] that integrates a microscopic simulation tool (VISSIM) with a robotics simulation platform and a discrete event simulation engine.

However, to the best of our knowledge there is no prior work focusing on the coupling of an open-source microscopic simulation framework with an ADAS oriented nanoscopic simulation tool. An interface between a traffic simulation environment (SUMO) and an ADAS environment (SCANeR) has been designed to make them consistent in the local area the ADAS simulator is evolving. As a result, the ego and surrounding vehicles evolution will be part of a highly frequently updated closed-loop system between both environments.

3. Simulation Environments' Integration

3.1 SCANeR and SUMO Overview

SCANeRTMstudio [21] is a driving simulation software package developed by OKTAL, based on former work of the Vehicles Simulation and Perception Research group by Renault. It is used for vehicle ergonomics, advanced engineering studies, road traffic research and development as well as for human factor studies and driver training.

The package includes several modules or modes [22]. The most relevant ones are:

- The *Terrain Mode*, which is designed to build a road network for driving simulation applications, based on a high level description.
- The *Vehicle Mode*, which is the dynamic part of SCANeRTMstudio software that analyses the behaviour of vehicles and pedestrians, in 3 dimensions, coupled and non-linear.
- The sensors package, containing models and tools for sensor simulation within the environment. Some available models are: radar, cameras, LIDAR, GPS, ultrasonic or road sensors.

SCANeRTMstudio is dedicated to single-computer or multi-computer simulations. Applications are built for 32-bit and 64-bit computer platforms. These applications use a common communication protocol. An Ethernet Network is used to transfer messages between applications. All platforms involved in the SCANeRTMstudio simulation must be linked to a Gigabit network. The following figure illustrates SCANeRTMstudio distributed architecture concept [22].

Moreover, SCANeR is an open software platform that provides developers with the ability to develop their own modules which will communicate with the rest of the simulation system without having to modify the core of the code source or to connect their personal software or hardware. To do that, the package includes a set of different APIs, most of them available in C/C++, python, c#/.net, Matlab Simulink and Labview. Each API is dedicated to a specific area of the simulation such as to create custom SCANeR processes, pilot the vehicle, create or modify the scenario, create a custom vehicle model or to integrate an external already existing model, or retrieve a stream from a camera sensor process for post processing.

SUMO (Simulation of Urban Mobility) [4] is an open source, highly portable, microscopic, multi-modal traffic simulation which allows to simulate how a given traffic demand which consists of single vehicles moves through a given road network. It is mainly developed by employees of the Institute of Transportation Systems at the German Aerospace Centre, and it is licensed under the GPL.

In a few words, using SUMO provides the following relevant advantages to this work:

- It is a purely microscopic traffic simulation where each vehicle is given explicitly, defined at least by an identifier (name), the departure time, and the vehicle's route through the network.
- It allows to simulate different types of vehicles.
- The simulation is space-continuous and time-discrete with a default simulation step length of 1s.
- It allows a fast execution speed (up to 100.000 vehicle updates/s on a 1GHz machine).
- It enables interoperability with other application at run-time.
- It allows to import many road network formats like OpenStreetMap, VISUM, Vissim, or MATsim.

However, such advantages are not enough for the intended purpose and additional features have been needed. So, different extensions have already been introduced:

- Plug-in to allow a TCP communication, and therefore, to receive information generated by other simulator at run-time.
- Adaptation of a theoretical model for fully electric vehicles. SUMO introduces the concept of "vehicle devices", such as an emissions control device or a person device for advanced person interchange, e.g. from private vehicle to bike or bus. Therefore, an additional "vehicle device" for fully electric vehicles has already been created to manage the simulation of common battery measures, and features such as efficiency ratios and weight and torque parameterization; the calculation of energy consumption by car, by area or by time; the consideration of car electric auxiliaries like lights, air conditioner, heating, etc; among others.
- Extension of the theoretical model for fully electric vehicles to highly automated vehicles. Another "vehicle device" is currently being created to manage the simulation of the specific information provided by the sensors on board the vehicle.

3.2 Communication and synchronization

To integrate and synchronize these two different-granularity simulators in the scope of automated and connected driving, a proper interface between the realistic nanoscopic simulator for ADAS (SCANeR) and the microscopic traffic simulator (SUMO) has been developed. A two-fold communication interface between both simulators deployed under Microsoft Windows Vista 7/8 OS and developed in C++ under Microsoft Visual Studio C++ with the purpose of, on the one hand, making use of the SCANeR API to communicate with it, and on the other hand, integrating it as a plugin within the code base of SUMO.

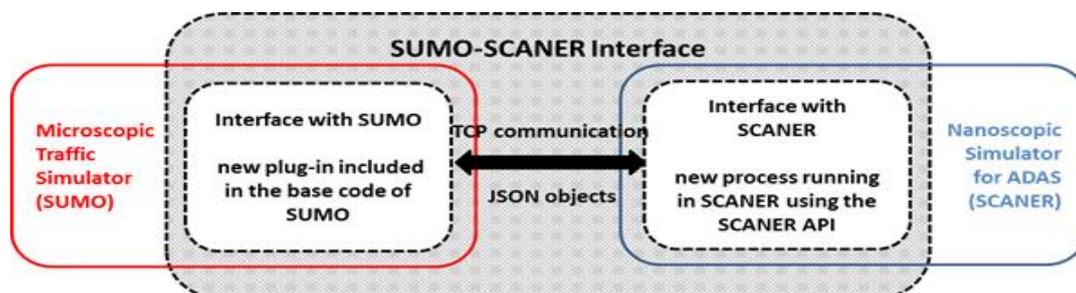


Figure 1. SUMO-SCANeR Interface at technical level

As observed in the figure 1, the communication between the two folds of the interface is a simple TCP communication where JSON objects are interchanged in both sides.

The interface with SUMO runs as a new plug-in integrated in the base code of SUMO. This plug-in is made up by the following relevant functions:

- Processing of requests from SCANeR, such as creating a new vehicle, updating a vehicle in SUMO by setting its speed/acceleration or changing the position of a given vehicle.
- Creation of requests to SCANeR, such as getting the bounding box in SCANeR, and getting the status data for a given vehicle.
- Transformation of Cartesian coordinates (SCANeR) into geographical coordinates (SUMO) for values related to the position and the route of the vehicles.

Moreover, the interface with SCANeR runs as a new SCANeR process using its API [23]. This API is delivered as a DLL with a "C language" binding, and it can be mapped with applications written in C, C++, C# or any application able to use a DLL. This dynamic library is a grouping of all available APIs with a common way to use it:

- *Process Controller* for event/state management for a process.
- *Simulation Controller* for supervising.
- *Communication Controller* for getting or sending data over network.

Thereby, the following actions can be done with this API:

- Connecting SCANeR™studio to external applications (LabView, Rapid++, MatLab/Simulink, etc), or devices in the simulation loop.
- Developing new SCANeR™studio modules in C or C++, such as an acquisition module, a data displayer module, or control autonomous vehicles, or more basically, send and receive numerical values through channels.
- Creating personalized launchers to manage these modules.

In order to develop a compliant interface with SCANeR, such interface must implement the following structure for the code:

```
int main(int argc, char* argv[])
{
    /*! Initialization of the Process */
    Process_Init(argc, argv);
    /*! Process State */
    APIProcessState status;

    do
    {
        /*! Wait for synchronisation at the given frequency */
        Process_Wait();

        /*! Run the process (and adapte the status) */
        status = Process_Run();

        if (status == PS_RUNNING)
        {
            // Add user code
        }
    } while (status != PS_DEAD);

    Process_Close();
}
```

Figure 2. Base code for any compliant SCANeR module

- *Process_Init* allows initializing the new SCANer process with command line parameters (behaviour of SCANer modules).
- *Process_Wait* allows sleeping the process. This function is used to schedule the client application of the SCANer API, and the frequency is given during the initialization.
- *Process_Run* allows running the process and launching all background tasks of the SCANer API. This function has to be called regularly by the client application to get the current state of the process.
- *Process_Close* allows closing properly the process.

Apart from this, it is also needed to define the data to be gotten/set and how such data is going to be exchanged. For this, it has been implemented a communication process as follows:

1. Declare the interface for data exchange with SCANer as input or output. In our particular case, it is used the *IVehicle* interface to get/set specific data about the vehicle.

Example:

```
DataInterface* vhcUpdate0 =
Com_declareInputData (NETWORK_IVEHICLE_VEHICLEUPDATE, 0);
```

2. Read/set data. In our particular case, it is requested the absolute position and the speed of the ego vehicle's CoG frame in the chassis referential.

Example:

```
Com_getFloatData(vhcUpdate0, "COGPos[0]")
Com_getFloatData(vhcUpdate0, "COGPos[1]")
Com_getFloatData(vhcUpdate0, "COGPos[2]")
Com_getFloatData(vhcUpdate0, "speed[0]")
```

Where [0], [1] and [2] correspond to x, y, z axes.

Once this interface running as a SCANer process is added to the Configuration Manager, then it has the same state management as the existing processes.

This interface between both simulators aims at making them consistent in the local area that the nanoscopic simulator is operating, as it is shown in figure 3.

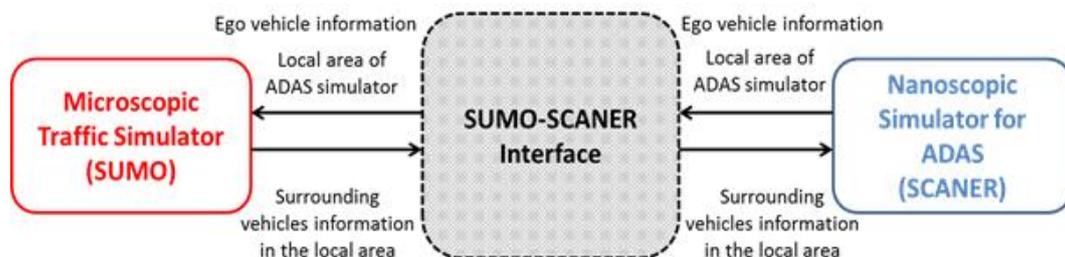


Figure 3. Flow of information through the SUMO-SCANer Interface

As a result, the evolution of the autonomous vehicle (the ego vehicle simulated by SCANer) and the surrounding vehicles in the ego vehicle's local area is part of a frequently updated closed-loop system between both environments.

Specifically, the synchronization between both environments through this interface is carried out for a given period, being of the order of 10 milliseconds, as it is shown in the following diagram:

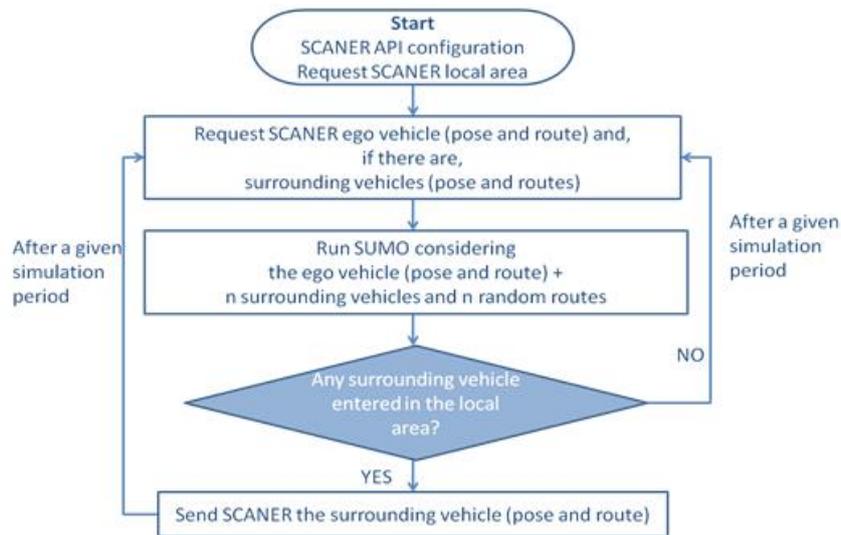


Figure 4. Synchronization through the SUMO-SCANer Interface

4. Use Case

Within the framework of the European project EMC2, a highly automated driving scenario has been demonstrated in the facilities of the Centre for Automation and Robotics (CAR) of the Spanish National Research Council. There, a ground test circuit has been designed as an inner city area, with a combination of straight-road segments, curves, 90 crossings and a roundabout, with V2X infrastructure and a traffic light regulation system.

Even though the facilities at CAR are very relevant for highly automated vehicles, it has not been at all easy to reproduce the increasingly complex driving situations with highly automated vehicles in a reduced and constrained experimentation site.

First, the road network comprising the ground test circuit at the CAR has been replicated, from real data, in our nanoscopic simulator SCANer, see figure 6. An ego-vehicle model, equipped with some sensors for autonomous driving such as stereo cameras and lidars, has been introduced in the environment. Also several other vehicles and pedestrians with automatic behaviours, to create different driving situations, and some 3D objects such as buildings and street furniture to replicate a realistic urban-like environment.

Within SUMO, we have created a much larger testing area, also from real data, containing the local area simulated within SCANer, see figure 5. In this much larger area, hundreds of vehicles have been included, each one following its own trajectory. Therefore, there is a portion of the whole map overlapped in both simulators, which is the one subject to synchronization.

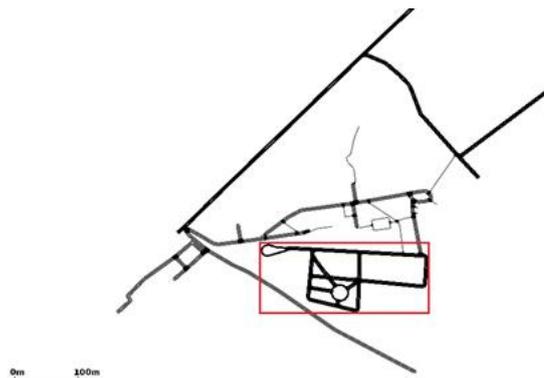


Figure 5. Working area for SUMO and local area for SCANer

By means of the communications interface described in the previous section, both simulators exchange data at all times. SCANer sends to SUMO the position and velocity of every vehicle simulated within its environment. This information is used in SUMO to update those vehicles within its own environment. Whenever a vehicle in SUMO enters the overlapped area, SCANer is commanded through the designed interface to create a new vehicle with a given itinerary. In the same way, whenever a vehicle leaves the simulated area in SCANer, SUMO is informed to take control of that vehicle so the simulation continues.

This allows the testing of functionalities at different levels. In one hand, the embedded software in charge of autonomous simulation of the ego-vehicle thanks to the SCANer capabilities to simulate the dynamics of the vehicle and include the detailed sensor models necessary for the vehicle's autonomous behaviour. However, this level of details can only be achieved simulating a limited number of vehicles in a relatively small area. On the other hand, SUMO is able to simulate a much larger area and number of vehicles at a higher level, where the vehicles' dynamics and sensor data is not necessary, but it is crucial to test applications like Intelligent Parking Management or Dynamic Ridesharing, where it is necessary having multiple drivers offering rides at different locations within a relatively large city as well as multiple potential users requesting for a ride.

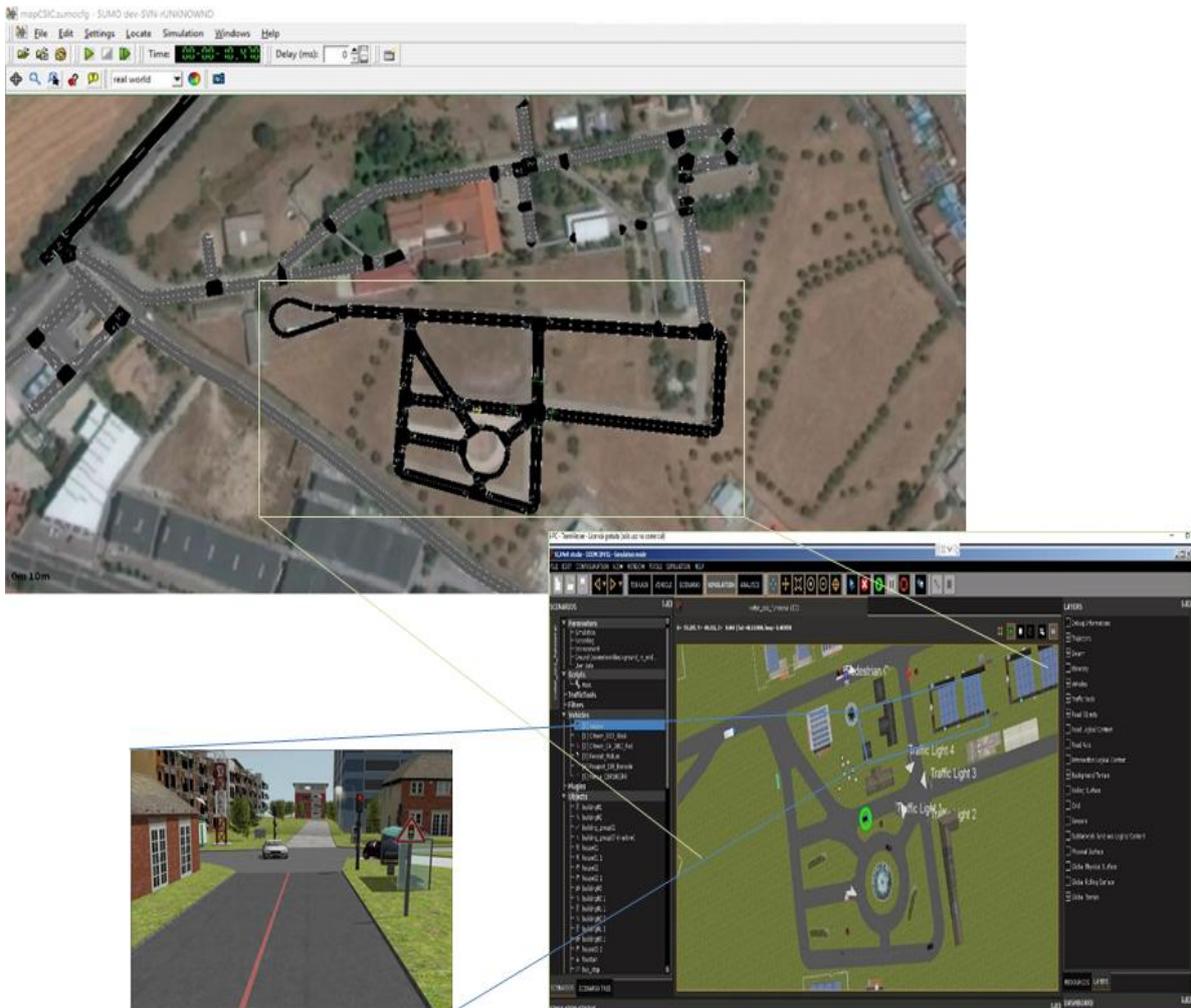


Figure 6. Simulation environments: (top) SUMO view from above, (bottom left) SCANer view from car's camera, (bottom right) SCANer view from above

5. Conclusions

The integration and synchronization of simulators of different granularity has been achieved by the design and development of a communications interface between SUMO and SCANeR. It allows combining together the features of a microscopic simulation framework with an ADAS-oriented nanoscopic simulation tool.

A use case to test this interface has been designed, where a given real urban area has been simulated at a high-level in SUMO, and a portion of this also simulated in SCANeR at a lower level, exchanging data through this interface to synchronize vehicles in both ends.

This light interface has allowed to seamlessly integrate both simulation tools with a relatively low effort and complexity. It has been verified how vehicles are actually synchronized in both ends in real-time, exposing the same behavior. Through this experiment, we have managed to test our embedded software for autonomous driving, which only interacts with SCANeR, while having the possibility to deploy hundreds of surrounding vehicles and users at a higher level to add high-potential capabilities to our ego autonomous vehicle like Intelligent Parking Management or Dynamic Ridesharing.

Acknowledgment

This work has been performed under project EMC2, which has received research funding from the EU ARTEMIS Joint Undertaking under Grant Agreement n° 621429 (EMC2), and from the Ministry of Industry, Energy and Tourism of Spain (MINETUR). Authors thanks EMC2 Consortium their cooperation along the lifecycle of the project.

References

- [1] Jimenez, F. et al, Intelligent Vehicles, Elsevier, 2017
- [2] Semrau, M. and Erdmann, J., 2016. Simulation framework for testing ADAS in Chinese traffic situations. SUMO 2016–Traffic, Mobility, and Logistics, 30, pp.103-115.
- [3] Ni, D., 2006, December. A framework for new generation transportation simulation. In Proceedings of the 38th conference on Winter simulation (pp. 1508-1514). Winter Simulation Conference.
- [4] Simulation for Urban Mobility - Wiki. [Online]. Available: http://sumo.dlr.de/wiki/Simulation_of_Urban_MObility_-_Wiki
- [5] Henderson, T.R., Lacage, M., Riley, G.F., Dowell, C. and Koppena, J., 2008. Network simulations with the ns-3 simulator. SIGCOMM demonstration, 14(14), p.527.
- [6] Hiblot, N., Gruyer, D., Barreiro, J.S. and Monnier, B., 2010. Pro-SiVIC and ROADS. A Software suite for sensors simulation and virtual prototyping of ADAS. In Proceedings of DSC, pp. 277-288.
- [7] <https://www.tassinternational.com/prescan>
- [8] <http://www.oktal.fr/en/automotive/range-of-simulators/software>
- [9] Carpin, S., Lewis, M., Wang, J., Balakirsky, S. and Scrapper, C., 2007, April. USARSim: a robot simulator for research and education. In Robotics and Automation, 2007 IEEE International Conference on (pp. 1400-1405). IEEE.
- [10] Michel, O., 1998, July. Webots: Symbiosis between virtual and real mobile robots. In International Conference on Virtual Worlds (pp. 254-263). Springer, Berlin, Heidelberg.
- [11] Tanaka, S., Mochida, T., Aga, M. and Tajima, J., 2012. Benefit estimation of a lane departure warning system using ASSTREET. SAE International Journal of Passenger Cars-Electronic and Electrical Systems, 5(2012-01-0289), pp.133-145.
- [12] Schiller, M., Dupuis, M., Krajzewicz, D., Kern, A. and Knoll, A., 2015. Multi-Resolution Traffic Simulation for Large Scale High Fidelity Evaluation of VANET Applications. SUMO 2015–Intermodal Simulation for Intermodal Transport, 28, pp.37-50.

- [13] Killat, M., Schmidt-Eisenlohr, F., Göbel, G., Kosch, T. and Hartenstein, H., 2007, March. On the accuracy of coupling a mobility and a communication simulator for VANETs. In Proc. 4th International Workshop on Intelligent Transportation (WIT), pp. 137-142.
- [14] Fellendorf, M., 1994, October. VISSIM: A microscopic simulation tool to evaluate actuated signal control including bus priority. In 64th Institute of Transportation Engineers Annual Meeting (pp. 1-9). Springer.
- [15] Schünemann, B., 2011. V2X simulation runtime infrastructure VSimRTI: An assessment tool to design smart traffic management systems. *Computer Networks*, 55(14), pp.3189-3198.
- [16] van Noort, M., Van Arem, B. and Park, B., 2010, Mobysim: An integrated traffic simulation platform. In 13th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 1301-1306.
- [17] Choffnes, D.R. and Bustamante, F.E., 2005, September. An integrated mobility and traffic model for vehicular wireless networks. In Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks, pp. 69-78
- [18] Potuzak, T., 2013, July. Issues of parallel hybrid nanoscopic/microscopic road traffic simulation. In EUROCON, 2013 IEEE (pp. 614-621). IEEE.
- [19] Keus, D., van Veldhoven, E.R., Heemskerck Pillis, F.V., Brink, T. and Laar, J.V., 2011. The Reality Check: Evacuation Planning done by Mixed Reality and Simulation. Simulation Interoperability Standards Organization's (SISO).
- [20] O'Hara, N., Slot, M., Marinescu, D., Čurn, J., Yang, D., Asplund, M., Bourroche, M., Clarke, S. and Cahill, V., 2012. MDDSVsim: an integrated traffic simulation platform for autonomous vehicle research.
- [21] ScannerTMstudio. [Online]. Available: <http://www.oktal.fr/en/automotive/range-of-simulators/software>
- [22] SCANeR™studio User Manual. Version 1.6. Oktal. June, 2016
- [23] SCANeR™studio SDK. Version 1.6. Oktal. June, 2016