Open-Source Integrated Simulation Framework for Cooperative Autonomous Vehicles

Christos Anagnostopoulos, Christos Koulamas, Aris Lalos and Chrysostomos Stylios
Industrial Systems Institute / Athena Research Center
Patras, Greece
Emails: anagnostopoulos@isi.gr, koulamas@isi.gr, lalos@isi.gr, stylios@isi.gr

***Abstract*—While many mature and reliable simulation engines exist for different system components, such as the network or the sensing and physics subsystems, the integration of these simulators into a single framework and workflow is not always a straightforward task. This paper presents such an integrated simulation framework, built from available open source components, focusing on cooperative autonomous vehicles research and development. The framework combines the CARLA simulation engine with the ETSI ITS-G5 implementation of the Artery/OMNET++ network simulator, as well as with selected functionality available in ROS, in order to provide a complete and high-fidelity integrated simulation environment which can be exploited for the development and testing of unaltered final code running in ROS-enabled devices.**

***Index Terms*—integrated simulator, CAV, V2X**

I. Introduction

The engineering of complex Cyber-physical systems and systems of systems requires a wide set of methods and tools, including the extensive use of simulations to develop and test the various aspects of the system. Research on autonomous vehicles encounters difficulties due to factors like the safety of people, the cost of the equipment, the availability of the vehicle, or the ability to control environmental conditions, factors which often become insurmountable barriers. Even though real-world testing is not fully replaceable, simulation can act supplementary and can accelerate the process since it can provide reproducible and scalable results on minimum cost.

The main objective of this work is to deploy a solution that combines a detailed network level simulator, such as OMNET++ with the CARLA emulator and the ROS framework, targeting to a platform that will support a distributed execution and testing of unaltered final code running in ROS-enabled devices interacting with a high-fidelity emulation of the physical environment. In this way, the system provides for a concurrent exploitation of highly tested and reliable wireless medium and network level models, as well as, emulations of sensors and processes related to cooperative autonomous vehicles, while the code that will be finally executed in the actual system remains unaltered.

The structure of the paper has as follows: in Section II, the various existing simulation solutions are presented and Section III describes the architectural and implementation aspects for the integration of the chosen subsystems. Section IV provides details from a specific simulation scenario and use case, while Section V concludes the paper.

This paper has received funding from the European Union's H2020 research and innovation programme CPSoSAware under grant agreement No 871738.

II. Related Work

In this section, the state of the art in simulation solutions is presented, targeting to all different aspects of the needed functionality, as well as to other integrated proposals.

A. Traffic Simulators

Traffic Simulators constitute an indispensable and quite mature tool for evaluating complex traffic systems since they can emulate the time variability of traffic phenomena [1]. Two of the most popular commercial simulators are PTV Vissim [2], which was first launched in 1992 and Aimsun [3]. Both are based on microscopic simulation meaning that each entity is simulated individually, and both can provide statistical results on complex implemented traffic scenarios. SUMO [4] is probably the most popular traffic simulator in research community since it's open source. In addition, it is compatible with formats form other simulators, like Vissim and it provides an API called TraCI which enables the bidirectional communication with other applications something that makes SUMO a suitable candidate for becoming a building block of an integrated simulator. These simulators can perform complex traffic scenarios but by themselves are not able to simulate physics events which makes them insufficient in simulating and connected autonomous vehicles' scenarios

B. Network Simulators

Network simulators in contrast to Traffic and Game Engine-based simulators can model and simulate properly the communications between the simulation entities. The main method utilized is simulations based on discrete events in which the simulation behaviors are not dictated by continuous equations but rather by temporal discrete events. One of the most popular simulators is ns-3 [5] which firstly released in 2008 offers an extensive set of libraries for modeling communication channels and protocols, but still lacks in features simulation vehicular networks. OMNeT++ [6] is a free, open-source, extensible, modular, component-based C++ simulation framework which also provides an Integrated Development Environment based on Eclipse. Independent frameworks extend OMNeT++ and provide support on real-time simulation, protocol implementations and network emulation. One extension like this is Artery [7] which it started as an extension of Veins framework but can now be used independently. Artery models and implements

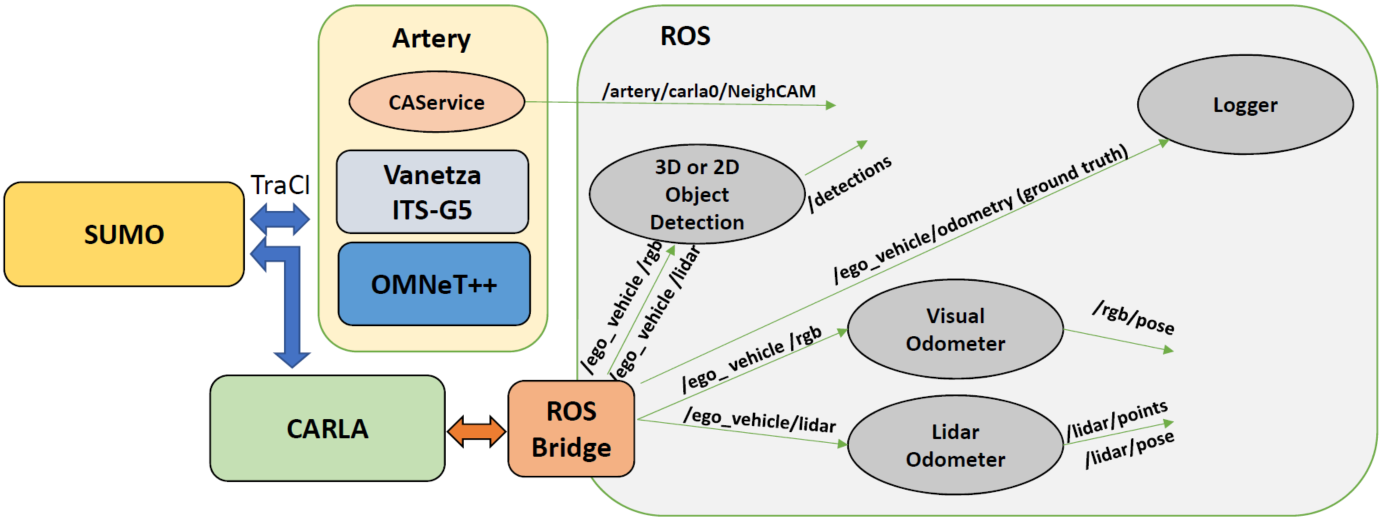

Fig. 1. Integrated Simulator Architecture

V2X simulations based on ETSI ITS-G5 protocols and facilitates services, such as Cooperative Awareness (CAMs) and Decentralized Notification (DENMs).

C. Game Engine-based Simulators

A virtual environment based on a game engine offers many comparative advantages over running experiments in the real world since there are far fewer restrictions in areas such as security, costs and control of environmental conditions. Key characteristics of a game engine like the rendering system, the physics engine and the ability to extent the functionalities of the components via scripting favor the development and the execution of autonomous vehicle scenarios.

CARLA [8] is probably the most popular game engine-based simulator amongst the research community. It is open source and based on the Unreal Engine [9]. It implements a scalable client-server architecture in which all simulation-related task are assigned to the server and the clients can control the simulation environment and entities via a Python API. SVL [10] is another end-to-end simulator platform built for the development and testing of self-driving vehicles, which is based on Unity3D [11]. Robot simulators like Gazebo [12] can also be utilized for simulating autonomous vehicles. Gazebo is a modular, open source simulator which can uses Open Dynamics Engine (ODE) as its default physics engine but can support other implementation too.

As pointed out in [13] the main problem with game engine-based simulators is that they may not be high fidelity. When a high-fidelity simulation, is required you may need to utilize an external component which contains a relevant mathematical representation to obtain realistic calculations.

D. Integrated Simulators

Even though, various simulation tools and frameworks exist in the area of autonomous vehicles, in many cases it is important to follow a multi-disciplinary approach, addressing the problem of simulating connected and autonomous vehicles in a holistic way covering all the aspects regarding the simulation of communications, the vehicle's behaviors, and V2X communications. Therefore, one obvious solution is the development of Integrated Simulators which combine the functionalities of tools already implemented. A very popular combination is the coupling of a network and a traffic simulator in which both ends run the same simulation and the synchronisation of their states is based on the information exchanged between the two parts. Following this approach implementations like ExNS3 [14],Veins [15] and iTETRIS [16] implement the aforementioned paradigm. In all cases a network simulator, OMMNeT++ or ns-3, and a traffic simulator exchange information via dedicated communication modules. However, this setup lacks data generated by physics engine and therefore is insufficient for testing scenarios that include for example perception algorithms.

Integrated simulators for evaluating a Cooperative Eco-driving System is presented in [17] and [18]. They model vehicular networks, driver models, vehicle models and control algorithms by integrating a network, a traffic, and a driving simulator. Both implementations do not have the ability to simulate physical phenomena since there is not a physics engine-based simulator. Finally, the work in [19] integrated Gazebo with Artery for supporting cooperative scenarios such as platooning.

III. Architecture and Implementation

The principal architectural detail refers to the combination and synchronization of the different clocks of the various subsystems of the combined simulation framework.

A. Components

Our framework combines three sub-systems, a network simulator, a traffic simulator and a game engine-based simulator, into a single platform.More specifically, we have selected CARLA as the component responsible for simulating physics phenomena and rendering. Artery V2X Simulation framework, which is built on top of OMNET++ framework, was our choice for simulating network communications and more specifically V2X communications [7]. For the control and the coordination of the simulating entities we have chosen SUMO.

B. Synchronisation and Orchestration

The CARLA simulator supports a set of four combinations of simulation step (fixed and variable) and client-server synchronicity (synchronous and asynchronous), among which, the simulation stability and results repeatability is achieved through the choice of a fixed time step and the synchronous client-server interaction mode. This allows a single external client to define the pace of simulation progress without any concerns regarding the processing speed mismatches. Since we have multiple sub-systems with their own stepping logic, it is evident that a single place must exist, that will act as a clock gate and synchronization point. This role can be realized by the Traffic Control Interface (TraCI) of the SUMO simulator that already supports interactions with both CARLA and OMNET++, in different contexts - in particular, with the Artery V2X Simulation framework, according to the Artery architecture .

Furthermore, CARLA interacts with ROS through the CARLA-ROS Bridge. Since in synchronous mode, only one client can tick the CARLA server, the Bridge must be also launched in passive mode, for the timing of the ROS subsystem to follow the single system clock source, too. Finally, in order to export to the ROS subsystem important, application level information, such as the ETSI ITS CAM or neighborhood from the Artery/OMNET++ network simulation, the ros-etsi-its-messages [20] encapsulation library can be used. For this, the artery CA service has been instrumented to provide an efficient dissemination of the current snapshot of the constructed neighborhood table related to each CARLA vehicle ID, as built from its own ITS CAM process, which is the ultimate abstraction needed at the ROS application code level.

The overall framework synthesis appears in Fig 1. The whole chain step is controlled by the slowest element, which is the network simulator and can be started or stopped through the ONMET++ user interface.

C. Execution

The orchestration of the simulation scenario execution is conducted by a Jenkins script and for that reason a REST API was developed. Jenkins is an open source continuous integration/continuous delivery and deployment (CI/CD) automation server.

After the initialization of the SUMO simulation server, the ego vehicle and other actors are spawned in CARLA. A central synchronisation entity is responsible for the coordination of the entities between the simulation environments. Artery via the TraCI interface acquires all the relevant data about the position and the routes of the vehicles. The simulation is controlled by Artery which ticks after the completion of every discrete step and publishes the CAM messages from the neighbouring vehicles of the ego under the ROS topic /artery/carla0/NeighCAM. CARLA-ROS bridge also publishes sensor data to the ROS environment and more specifically, rgb images, lidar data and the ground truth position of the ego vehicle. The odometry algorithms subscribe to the relevant topics, depending on their input modality, and produce estimated poses and point clouds which are published under the corresponding topics. Finally, a logger ROS node is used for capturing all the outputs of the algorithms along with the ground truth acquired directly from CARLA.

IV. Simulation Scenario and Use Case

Fig. 2 demonstrates the whole setup of the simulation framework and how the object detection, multi-modal fusion and cooperative localization algorithms are integrated.

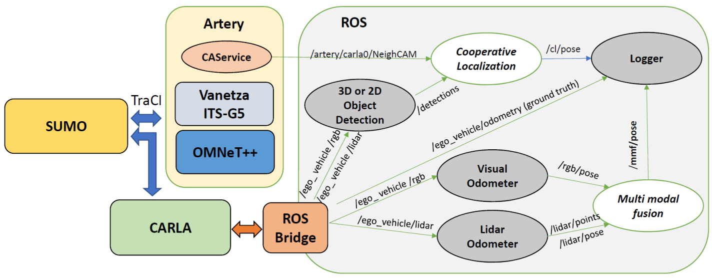

Fig. 2. Integrated Simulation Scenario

The multi-modal fusion ROS node implements a fusion algorithm which utilises the output of the odometry algorithms aiming at the robustification of the ego vehicle's pose estimation. The cooperative localization ROS node uses the CAM messages from Artery and in combination with the output of the object detector node estimates the position of the ego-vehicle. The algorithmic details of this functionality block have been described in [21]

V. Conclusions and Future Work

This paper presents a simulation framework that integrates seamlessly a network, a traffic and a game engine-based simulator for evaluating CAV systems. We have executed a simulation scenario for validating our setup and demonstrating how this implementation will benefit the research community since it can run complex traffic scenarios on varying environmental conditions and it generates synchronised synthetic data that include highly realistic sensor data and V2X messages compliant to ETSI ITS-G5 protocols. In the future, we plan to build a dataset that will contain synchronized data from both the physics and the network simulator encapsulated in a rosbag. This will become a valuable tool for evaluating implementations and algorithms like the aforementioned example of the cooperative localization ROS node which combines the V2X messages from

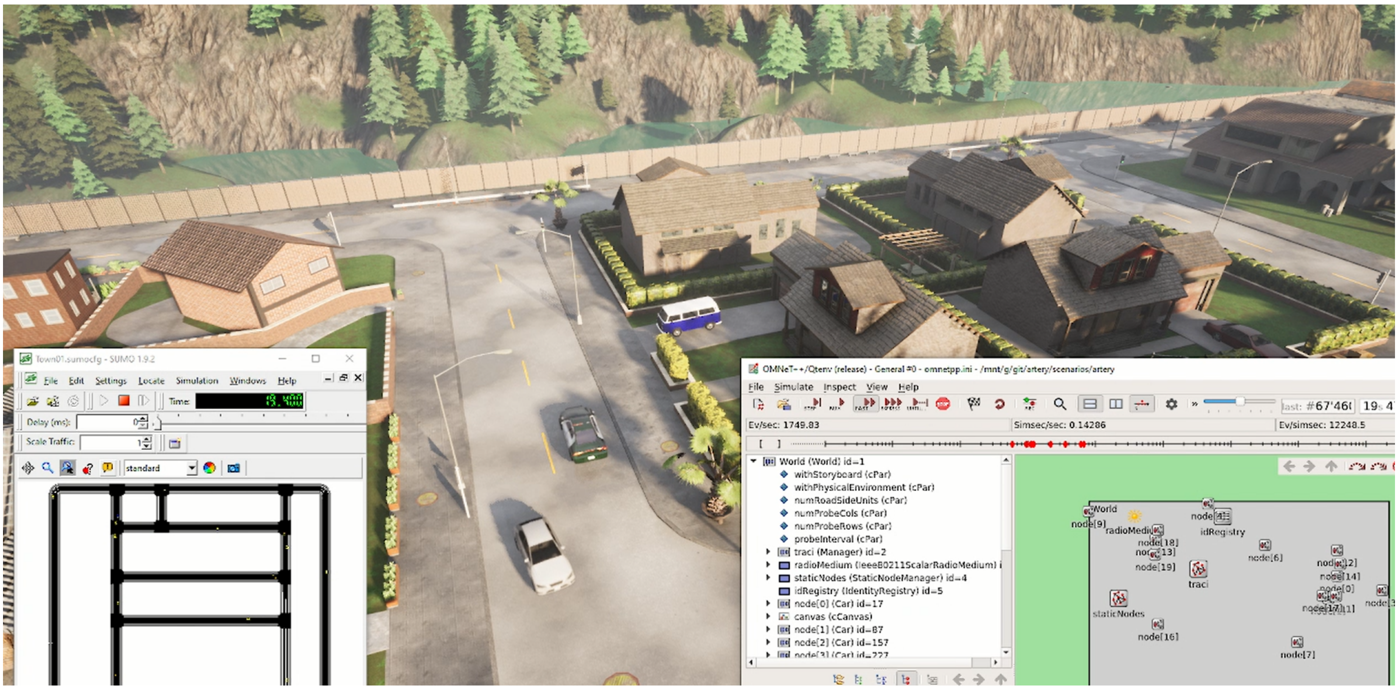

Fig. 3. Snapshot of an integrated simulation GUI

Artery and the output of an object detector node to estimate the position of the ego vehicle.

REFERENCES

[1] J. Barceló, Ed., *Fundamentals of Traffic Simulation*. Springer, 2010. [Online]. Available: https://EconPapers.repec.org/RePEc:spr:isorms:978-1-4419-6142-6

[2] "Traffic simulation software — ptv vissim — ptv group," https://www.ptvgroup.com/en/solutions/products/ptv-vissim/, (Accessed on 02/18/2022).

[3] "Aimsun: simulation and ai for intelligent mobility," https://www.aimsun.com/, (Accessed on 02/18/2022).

[4] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. [Online]. Available: https://elib.dlr.de/124092/

[5] "ns-3 — a discrete-event network simulator for internet systems," https://www.nsnam.org/, (Accessed on 02/20/2022).

[6] A. Varga, *OMNeT++*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 35–59. [Online]. Available: https://doi.org/10.1007/978-3-642-12331-3_3

[7] R. Riebl. Artery architecture. [Online]. Available: http://artery.v2x-research.eu/architecture

[8] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," 2017.

[9] "The most powerful real-time 3d creation tool - unreal engine," https://www.unrealengine.com/en-US/, (Accessed on 02/19/2022).

[10] "Svl simulator by lg - autonomous and robotics real-time sensor simulation, lidar, camera simulation for ros1, ros2, autoware, baidu apollo. perception, planning, localization, sil and hil simulation, open source and free." https://www.svlsimulator.com/, (Accessed on 02/20/2022).

[11] "Unity real-time development platform — 3d, 2d vr & ar engine," https://unity.com/, (Accessed on 02/19/2022).

[12] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, 2004, pp. 2149–2154 vol.3.

[13] F. Rosique, P. J. Navarro, C. Fernández, and A. Padilla, "A systematic review of perception system and simulators for autonomous vehicles research," *Sensors*, vol. 19, no. 3, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/3/648

[14] H. Arbabi and M. C. Weigle, "Highway mobility and vehicular ad-hoc networks in ns-3," 2010.

[15] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing (TMC)*, vol. 10, no. 1, pp. 3–15, January 2011.

[16] M. Rondinone, J. Maneros, D. Krajzewicz, R. Bauza, P. Cataldi, F. Hrizi, J. Gozalvez, V. Kumar, M. Röckl, L. Lin, O. Lazaro, J. Leguay, J. Härri, S. Vaz, Y. Lopez, M. Sepulcre, M. Wetterwald, R. Blokpoel, and F. Cartolano, "itetris: A modular simulation platform for the large scale evaluation of cooperative its applications," *Simulation Modelling Practice and Theory*, vol. 34, pp. 99–125, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569190X1300018X

[17] G. Lee and J.-I. Jung, "Integrated simulator for evaluating cooperative eco-driving system," in *2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, Dec 2019, pp. 1–2.

[18] Y. Zhao, A. Wagh, Y. Hou, K. Hulme, C. Qiao, and A. W. Sadek, "Integrated traffic-driving-networking simulator for the design of connected vehicle applications: Eco-signal case study," *Journal of Intelligent Transportation Systems*, vol. 20, no. 1, pp. 75–87, 2016, cyber Transportation Systems and Connected Vehicle Research. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1547245022000354

[19] B. Vieira, R. Severino, E. V. Filho, A. Koubaa, and E. Tovar, "Copadrive - a realistic simulation framework for cooperative autonomous driving applications," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, 2019, pp. 1–6.

[20] R. Riebl". ros-etsi-its-msgs. [Online]. Available: https://github.com/riebl/ros_etsi_its_msgs

[21] N. Piperigkos, A. S. Lalos, and K. Berberidis, "Multi-modal cooperative awareness of connected and automated vehicles in smart cities," in *2021 IEEE International Conference on Smart Internet of Things (SmartIoT)*, 2021, pp. 377–382.