Bridging Worlds: Integrating Hardware-in-the-Loop Testing with Large-Scale VANET Simulation

Dominik S. Buse*, Max Schettler*, Nils Kothe†, Peter Reinold†, Christoph Sommer* and Falko Dressler*
*Heinz Nixdorf Institute and Dept. of Computer Science, Paderborn University, Germany
†dSPACE GmbH, Paderborn, Germany
{buse,max.schettler,sommer,dressler}@ccs-labs.org
{nkothe,preinold}@dspace.de

***Abstract*—Modern vehicle systems are becoming more complex with every generation and development cycles are becoming ever shorter. Hardware-in-the-Loop (HIL) simulation is used as a key technology as it supports the development of specialized Electronic Control Units (ECUs) under realistic and reproducible conditions. It allows hardware prototypes to be tested with all of the rest of the car's functionality being simulated in real-time. At the same time, Car-to-X (C2X) communication is becoming a more and more vital component for next generation vehicles. Here, network simulation is used to explore the capabilities of the system – typically using a discrete event simulation approach. In this paper, we discuss the necessity to integrate simulators from both domains in order to assess complex interacting cooperative driving systems. In particular, we propose an approach for integration which we named Ego-Vehicle Interface (EVI) and we show first results that underline the feasibility of our concept.**

I. Introduction

Modern vehicle systems are becoming more complex every day and development cycles increasingly make use of simulation. In traditional Hardware-in-the-Loop (HIL) studies, typically a single vehicle (the *ego* vehicle) is simulated with the finest suitable granularity as far as time scale and model complexity are concerned [1], [2]. HIL studies in the context of Advanced Driver Assistance Systems (ADAS) also often model a select few *fellow* vehicles (i.e., vehicles surrounding the ego vehicle), that can perform simple interactions. As HIL simulators interact with real hardware, they are run in real-time.

In recent years, wireless communication has proven to enable cars to become part of a complex interacting system, a smart city [3]. To accurately simulate such interacting systems, the *context* of a car – the Vehicular Ad-Hoc Network (VANET) – must therefore be simulated. In VANET simulation, typically both the system under test *and* the context are modeled in a city-scale simulation. The VANET simulator takes care of simulating the mobility of all participants as well as the Car-to-X (C2X) communication between them and with infrastructure elements such as traffic lights [4]. VANET simulation is typically realized as a discrete event simulation running in a "best effort," non-real-time fashion with flexible time granularity – that is, time progress in the simulation (simulation time) is fully decoupled from real (or wall-clock) time. This allows simulation performance to scale with the size of the system under study and/or the required granularity of the results.

It is thus obvious that both simulation domains (HIL and VANET simulation) follow conflicting approaches and differ not just in model granularity and time granularity, but even in time progress (discrete-event vs. real-time). In this paper, we propose an approach, named Ego-Vehicle Interface (EVI), to integrate these very different types of simulators. As an example, we picked the dSPACE Automotive Simulation Models (ASM) (running on a dSPACE HIL simulator [5]) and Veins [6] (a state of the art VANET simulator). We describe how we extended both simulators with a real-time system interface coupling both simulation domains and we present first results.

Our key contributions can be summarized as follows:

- We discuss the necessity to integrate simulators of both real-time and non-real-time domains.
- We propose a conceptual approach, the Ego-Vehicle Interface, and discuss its implementation.
- Guided by first performance measurements, we are able to demonstrate the feasibility of the presented approach.

II. Related Work

Already in 2009, an approach has been proposed to provide traffic and other environmental data to various X-in-the-Loop systems, but lacking any C2X capabilities or notion of large-scale traffic [7]. More recently, a virtualization-based approach for emulating VANET-enabled ADAS implementations has been presented [8]. It allows to run the unmodified implementations of applications online on virtualized Electronic Control Units (ECUs) connected to both a vehicle simulator and a wireless network simulator. Time progress in this approach is synchronized live among simulators but not to wall-clock time, thus, prohibiting coupling actual real-time systems.

Instead of a HIL, the real-time system could also be a human-controlled device such as a driving simulator. This is a similar problem, as such simulators also can only manage a limited number of surrounding vehicles that interact with the ego vehicle. For this, a framework that dynamically selects nearby vehicles to synchronize to a networked driving simulator has been proposed [9]. While it allows a real-time system to be coupled to a potentially large-scale traffic simulation, it

ISBN 978-3-903176-02-7

is limited in terms of the road network layout and does not support any kind of C2X communication. There have also been attempts to build a three-in-one simulator combining 3D driving, traffic, and wireless simulation [10]. However, this offers only limited scenarios, no city-scale road networks, and it couples only soft real time components.

In another approach, a real car has been integrated into a traffic simulator [11]. The car is connected to a smartphone (via OBD-II), which in turn has a 3G UMTS uplink for data exchange over the Internet. The smartphone forwards the vehicle status updates to a server running a traffic simulator and acts as an interface to the driver by displaying ADAS information. This way, the authors were able to implement a demo speed-advisory system. However, while the car is coupled with the traffic simulation, messages are exchanged only every second and over a cellular network that is susceptible to delay and jitter, thus, not supporting hard real-time bounds.

One challenge of C2X-support is the complexity of computation necessary for simulating wireless communication. For this, the feasibility of a C2X simulator implementing the ETSI ITS G5 standard for providing data for a HIL has been analyzed [12]. However, the authors conclude that their setup is sufficient for only a few vehicles and only covers one-hop communication. In addition, the HIL is simulating only the wireless behavior of the ego vehicle in this approach.

In summary, to the best of our knowledge, we present the first solution coupling real-time HIL systems and large-scale VANET simulations.

III. EVI Concept

A. System Architecture

Conceptually, a system that integrates both real-time HIL and large-scale VANET simulation needs three components: the two simulation systems themselves and a coordinating interface. We describe these components in the following.

The real-time system can either be a pure HIL simulator or it can be a simulator which is coupled with an actual physical system like an engine. Its purpose is to provide real-world behavior (or a highly detailed simulation thereof) of a particular system, that could not be simulated by the VANET simulator. In any case, in terms of interaction, the real-time system always has to be treated like a physical system: It cannot change its state instantly in discrete steps but is doing so continuously. An example is an engine that cannot instantly change its rotational speed, even though a simulated artifact may pretend to do so. Interactions with the real-time system can, thus, only be done by means of stimulating sensor input of system or by means of input values to a systems internal control loop.

VANET simulators [4], e.g., Veins, VSimRTI, iTETRIS, NCTUns, or CityMoS, typically behave more like traditional computer programs running on a best-effort strategy: sometimes faster, sometimes slower, depending on the complexity of the task and system load. They are responsible for simulating both large-scale road traffic and C2X communication. A common approach is to achieve this by incorporating two separate simulators (or simulation engines) for wireless and traffic

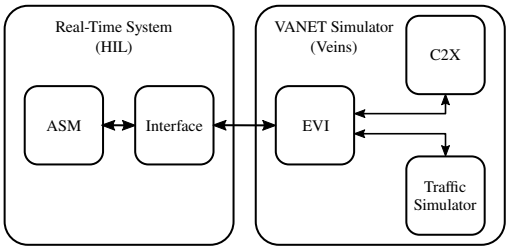

Figure 1. Architecture and data flows of the coupled simulation system.

simulation: The C2X simulator is taking care of all wireless communication activities. It controls the progress of simulation time while the traffic simulator acts as a slave that provides vehicle behavior updates on request.

The coupling of discrete event simulation with a real-time system is complicated as the progress of *simulation time* is then bound to the *wall clock time*. Thus, a new component is necessary that orchestrates the progress of simulation time in both the road traffic simulator and the C2X simulator, keeping it synchronized to the real-time clock.

For orchestration, we developed a new component that we named *Ego-Vehicle Interface (EVI)*. As depicted in Figure 1, it builds a bridge between the real-time system and the C2X simulator. By listening to clock tick messages from the real-time system, it triggers timed computations and the advance of simulation time within the non-real-time C2X and road traffic simulators. All exchanged messages flow through the EVI. This allows it to manage, filter, and time the data across the real-time bridge in order to react to deviations in real-time synchronicity. To minimize additional delays introduced by the new component, we designed the EVI around an asynchronous execution and networking model. This allowed us to integrate the components into a coupled system without large changes of the components themselves.

For our implementation, we chose ASM running on a HIL simulator as the real-time system and Veins as the C2X simulation component. ASM is a simulation tool suite for automotive controller development and test in all steps of a model-based design process. ASM supports simulation of automotive components in drivetrain, vehicle dynamics, and driver assistant applications. Veins consists of a C2X simulation component running in OMNeT++ connected to the large-scale traffic simulator Simulation of Urban Mobility (SUMO). Our system also provides information about selected fellow vehicles to the ASM environment so that driving functions can react upon their presence. In return, the ego vehicle information is fed back into Veins to update its position and speed in the road traffic simulator SUMO.

B. Data Flows

As shown in Figure 1, all data flows cross the EVI, which acts as an intelligent broker. It implements the concept of a Region of Interest (ROI) around the *ego vehicle* to select suitable vehicles to appear as *fellows* in the HIL, i.e., cars that driving functions might have to consider. Pre-computed results from the traffic simulation containing vehicle state updates are

ISBN 978-3-903176-02-7 © 2018 IFIP

cached by the EVI. When the wall clock time advances to their time of validity, the traffic, or a subset thereof, is forwarded to both the vehicle simulation on the HIL and C2X simulator.

The real-time system sends periodic messages including ego vehicle state updates. These messages are also considered as real-time clock ticks for synchronization by the EVI. In turn, the real-time system receives updates of selected fellow vehicles and macroscopic traffic information. The C2X simulator also receives vehicle status updates, but covers a larger area and greater number of vehicles to simulate. Depending on the implemented C2X protocols, it can send changes of non-ego vehicle behavior to be forwarded to the traffic simulation. Messages received by the ego vehicle can be forwarded to the HIL via the EVI. The road traffic simulator provides updates of the vehicle states to the EVI for further processing and distribution. Between simulation steps, it receives updates of the state of the ego vehicle and incorporates them into its own model. It also receives a *compute* command to trigger simulation of the next time step.

In addition to the data exchanged at runtime, the components need to share knowledge about the simulation scenario. Especially a model of the road network needs to be available to both the HIL and the traffic simulator. As such models are highly platform-dependent, suitable mapping tables between the native models are required. Both the model and the mapping tables are computed and shared offline.

C. Timing

There is a difference of two orders of magnitude in the time scales on which the different simulators operate. The traffic simulation computes updates of vehicle states for cycles (or intervals) of 100 ms. At this cycle time, the traffic simulator provides a good trade-off between model validity, computation effort, and result granularity. For the traffic simulator, these intervals are assumed to be one atomic step for all vehicles in the simulation. Modification commands from other simulators are only possible between the computations of two cycles.

OMNeT++ (which hosts the C2X simulation) strictly follows the discrete event simulation principle. By processing events, the reaction of the simulated model is computed, which in turn can schedule future events. Once an event is processed, it jumps to the next scheduled event by immediately advancing the simulation time to this event's time. In addition, it schedules events every 100 ms to receive traffic updates from SUMO. As events can occur at any moment in time, the C2X simulator has to extrapolate the position of vehicles. Furthermore, messages received by the ego vehicle are directly forwarded to the HIL via the EVI, without waiting for the next synchronization cycle.

In contrast, the HIL approximates continuous real-time behavior by using small cycles of 1 ms. The model on the HIL also extrapolates fellow vehicle positions between synchronization cycles with the EVI. As a real-time system, the HIL is assumed to be the only valid source of wall clock time. Thus, all other systems adjust to it by reacting upon the messages from the HIL. The entire process is again coordinated by the EVI.

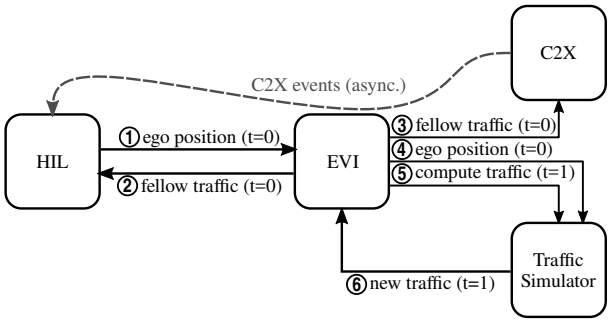

Figure 2. Flow and order of exchanged messages among simulator components. Synchronous message cycles (1 to 6) repeat every 100 ms. C2X event messages relevant for the ego vehicle are sent asynchronously in the moment they occur.

(a) Veins simulator (b) ASM real-time system

Figure 3. Running co-simulation with ego (blue) and fellow vehicles.

Figure 2 shows the order of messages that implement this principle. At the beginning of every synchronization cycle of 100 ms, the HIL sends an update message that is also considered a clock tick by the EVI. The EVI reacts by immediately replying with the cached and filtered traffic state provided by the traffic simulation in the last cycle. It then also sends a traffic update to the C2X simulator, which also contains the updated ego vehicle position received from the HIL. The traffic simulator, in turn, receives the updated ego vehicle state and the command to advance by one cycle. Once it is finished, it returns the updates to the EVI which caches and filters the data. Then the EVI waits for the next message from the HIL to start the next cycle.

IV. Performance

To demonstrate the feasibility of our concept, we implemented the EVI as a daemon that connects all three components of the system: ASM running on a dSPACE Scalexio HIL system, Veins C2X simulation running in OMNeT++, and SUMO performing traffic simulation. The HIL is connected via Ethernet to a Linux computer that runs all other components (including the EVI daemon). Over this Ethernet link the HIL and the EVI daemon communicate via UDP messages.

As an example scenario, the ego vehicle runs Adaptive Cruise Control (ACC), which controls the speed of the vehicle also taking the speed of *fellow* vehicle in front into account. A visualization of the resulting coupled simulation is shown in Figure 3 in the form of screenshots of the respective simulation components. Figure 3a shows a bird's eye view of the same scene in Veins and Figure 3b shows the ego vehicle as simulated by ASM, with some fellows simulated by SUMO. The road

ISBN 978-3-903176-02-7 © 2018 IFIP

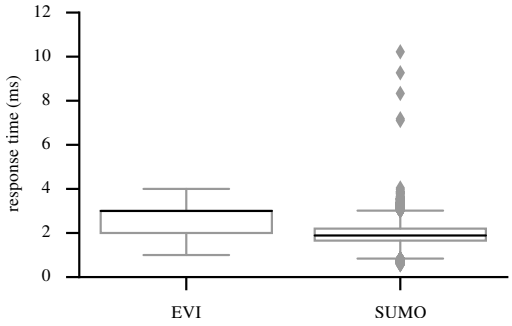

Figure 4. Response times for updated traffic data. The left depicts the number of 1 ms time steps ASM has to wait for updates from EVI. The right depicts the time SUMO needs to compute a time step and report back to EVI.

network for this scenario is a simple circle of a bi-directional road on which the ego vehicle can drive virtually indefinitely. Other vehicles are periodically added in the SUMO traffic simulation, driving both in the same and the opposite direction as the ego vehicle.[1]

A core metric of success for this scenario is whether the EVI can provide fellow traffic to ASM fast enough. Thus, we recorded the number of time steps in ASM between sending the clock tick message (which also contains the ego vehicle updates) to and receiving the traffic update message from the EVI. This period should be as low as possible or, even better, have an upper bound. Otherwise, the interpolations of the fellow vehicles produced by ASM would deviate too far from the values in the traffic simulation and could lead to incorrect ACC behavior. This measurement was recorded on the HIL to benefit from its real-time clock, which operates in cycles of 1 ms. Thus, measured response times are multiples of that.

For a second metric, we recorded the duration EVI has to wait while SUMO simulates a time step. In more communication-heavy scenarios, wireless communication will probably become the performance bottleneck. But for now, we want to ensure that traffic reliably provided. To this, duration has to stay below the synchronization cycle time of 100 ms. As long as this condition holds, the response time of SUMO is not visible to ASM, as traffic data is cached in the EVI.

The results of these measurements are shown in Figure 4, which depicts the distribution of the response time of EVI (as perceived by ASM) and SUMO (as perceived by the EVI). The maximum response time of the EVI lies at 4 ms and 75 % at or below 3 ms, thus, we can say that the response time stays bounded to an acceptable latency. We can also see that the computation time required by SUMO does not lead to delays in the response time of the EVI (as we would expect, thanks to caching traffic results in the EVI). The response time of SUMO also stays low enough in the simulated scenario. The median response time of about 2 ms shows that SUMO has no problems simulating the overall traffic in this scenario. Even the longest measured period of roughly 11 ms stays well below the cycle time of 100 ms. This leaves us with considerable reserves for data handling within the EVI or even larger scenarios.

[1]A video can be found at: http://www.hy-nets.de/media/evi-ring-road.mp4

V. Conclusion

With this paper, we aimed to bridge the domains of high-detail real-time and large-scale VANET simulation. We discussed the necessity to integrate such simulators into one coupled system, and then proposed a conceptual approach. Our system, which we named EVI, couples ASM running on a real-time HIL system with the VANET simulator Veins. To demonstrate its feasibility, we implemented the EVI and found its response time to be sufficiently short. With this concept as a foundation, we enable future research to harness combined benefits of highly detailed HIL testing embedded in large-scale VANET traffic scenarios. This can be used to analyze ADAS (such as, but not limited to, ACC) at unpreceded levels of detail and context. In future work, we want to extend the presented platform to support more complex scenarios, further improve its performance, and release it as open source.

Acknowledgements

Research reported in this paper was conducted in part in the context of the Hy-Nets project, supported by the European Regional Development Fund (EFRE-0800342).

References

[1] R. Isermann, J. Schaffnit, and S. Sinsel, “Hardware-in-the-loop simulation for the design and testing of engine-control systems,” *Elsevier Control Engineering Practice*, vol. 7, no. 5, pp. 643–653, May 1999.

[2] O. Gietelink, J. Ploeg, B. D. Schutter, and M. Verhaegen, “Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations,” *Taylor & Francis Vehicle System Dynamics*, vol. 44, no. 7, pp. 569–590, Feb. 2006.

[3] C. Sommer and F. Dressler, *Vehicular Networking*. Cambridge University Press, Nov. 2014.

[4] C. Sommer, J. Härri, F. Hrizi, B. Schünemann, and F. Dressler, “Simulation Tools and Techniques for Vehicular Communications and Applications,” in *Vehicular ad hoc Networks - Standards, Solutions, and Research*, C. Campolo, A. Molinaro, and R. Scopigno, Eds., Springer, May 2015, pp. 365–392.

[5] A. Himmler, “Hardware-in-the-Loop Technology Enabling Flexible Testing Processes,” in *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, Grapevine, Texas: AIAA, Jan. 2013.

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

[7] K. von Neumann-Cosel, M. Dupuis, and C. Weiss, “Virtual Test Drive – Provision of a Consistent ToolSet for [D,H,S,V]-in-the-Loop,” in *Driving Simulation Conference Europe*, Monaco, Feb. 2009.

[8] M. Schiller and A. Knoll, “Emulating Vehicular Ad hoc Networks for Evaluation and Testing of Automotive Embedded Systems,” in *8th EAI International Conference on Simulation Tools and Techniques (SIMUTools 2015)*, Athens, Greece: EAI, Aug. 2015, pp. 183–190.

[9] K. Abdelgawad, S. Henning, P. Biemelt, S. Gausemeier, and A. Trächtler, “Advanced Traffic Simulation Framework for Networked Driving Simulators,” in *8th IFAC Symposium on Advances in Automotive Control (AAC 2016)*, Norrköping, Sweden: Elsevier, Jun. 2016.

[10] Y. Hou, Y. Zhao, A. Wagh, L. Zhang, C. Qiao, K. F. Hulme, C. Wu, A. W. Sadek, and X. Liu, “Simulation-Based Testing and Evaluation Tools for Transportation Cyber-Physical Systems,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1098–1108, Mar. 2016.

[11] W. M. Griggs, R. H. Ordóñez-Hurtado, E. Crisostomi, F. Häusler, K. Massow, and R. N. Shorten, “A Large-Scale SUMO-Based Emulation Platform,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3050–3059, Dec. 2015.

[12] C. Obermaier and C. Facchi, “Observations on OMNeT++ Real-Time Behaviour,” in *4th OMNeT++ Community Summit (OMNeT++ 2017)*, arXiv:1709.02207v1, Bremen, Germany: arXiv, Sep. 2017.