2016 IEEE Intelligent Vehicles Symposium (IV), pp. 144-150, 19-22 June 2016.

Testing and Validating High Level Components for Automated Driving: Simulation Framework for Traffic Scenarios

Marc René Zofka, Sebastian Klemm, Florian Kuhnt, Thomas Schamm and J. Marius Zöllner

***Abstract*— Current advances in the research field of autonomous driving demand advanced simulation methods for testing and validation. By combining versatile foci of different simulations, we can provide an increased amount and diversity of realistic traffic scenarios, which are relevant to the development and verification of high level automated driving functions. The focus of the present paper is to propose a concept for realistic simulation scenarios, which is capable of running in different integration levels, from software- to vehicle-in-the-loop. Its application is demonstrated, exposing an experimental vehicle, which is used for autonomous driving development, to a traffic scenario with virtual vehicles on a real road network.**

I. INTRODUCTION

Since the DARPA's Grand Challenge in 2004 and Urban Challenge in 2007, huge progress has been made towards automated driving. Different research groups have demonstrated highly integrated automated driving functions, for example see [1], [2], [3], [4]. These systems go beyond advanced drivers assistance systems (ADAS), such as automated emergency braking assistance (AEB) by integrating task-specific, cognitive skills to cope with an increasing amount of traffic situations. Support for or even replacement of the driver promises a reduction of traffic accidents, a reduction of traffic congestions and thereby an energy-efficient way of mobility.

The evaluation and validation of automated driving functions become more and more complex as those systems' workspace increases. As a countermeasure the need for efficient and cost-effective possibilities to test those systems arises. Real road tests would require millions of kilometers in order to obtain a statistical relevant statement about the systems performance, being only valid for an exact mechanical configuration and algorithmical parametrization. Therefore, this way of validation and verification is not applicable for a development process, such as the V-model [5]. Automated driving functions have to be validated in a variety of complex traffic situations, including the uncertainty resulting from different cooperative traffic participants. Virtual assessment by simulation promises to provide a comfortable way to assess the performance of algorithms under an extensive variation of scenario parameters. Simulations can be used at different integration levels during the whole development process and facilitates repeatable test runs and traceable evaluation.

The authors are within Department of Technical Cognitive Assistance Systems, FZI Research Center for Information Technology, Karlsruhe, Germany, Email: {zofka, klemm, kuhnt, schamm, zoellner}@fzi.de

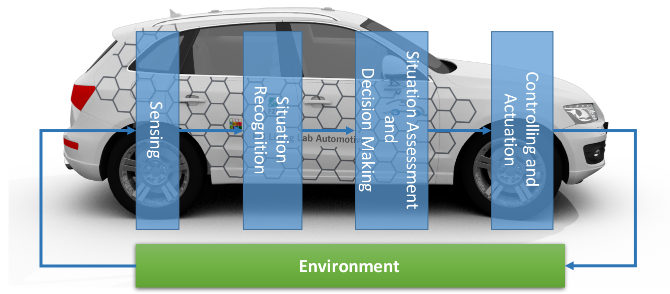


Fig. 1: Coarse scheme for automated driving functions.

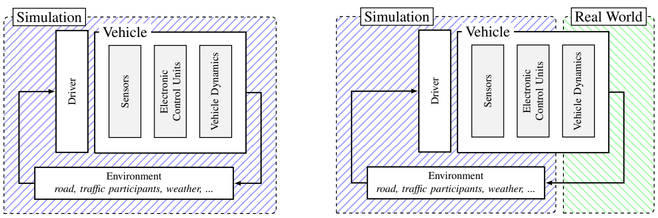
Automated handling of complex traffic situations, such as those exposed to a highway pilot or to an urban driving assistance, requires high level components, which take care of perceived and deduced environmental situation aspects. Typical components are the prediction of future traffic participants' trajectories and planning of appropriate trajectories. A coarse scheme for such tasks is depicted in Fig. 1. These components rely on different information from different sources, like online perception, offline maps or vehicle-to-vehicle communication. In order to verify such high level components, a simulation environment is necessary, which provides this information in a specific manner using noisy sensor models, as well as ground truth to evaluate the system's response.

In this paper, we present a testing framework, which is based on the co-simulation of a *vehicle mechanics and sensor simulation* with a *traffic flow simulation*. We demonstrate the application in different integration steps, from software- to vehicle-in-the-loop. The vehicle mechanics simulation serves as an integration interface as well as an object-oriented environment model, called scene graph model. This model provides ground truth as well as noisy information access through generalized sensor models, whereas the traffic simulation determines the dynamic behaviour such as those of traffic participants. Thereby, a structured approach for the variation of surrounding traffic given real world road networks for the virtual assessment of highly automated driving functions is achieved.

This paper is structured as follows: After motivating our approach, we give an overview of the related simulation concepts in Sec. II. Then, in Sec. III the concept for the simulation framework is described in detail and in Sec. IV the realization is presented. We demonstrate the application of our framework in Sec. V. Finally, we discuss our results and illustrate future work in Sec. VI.

II. RELATED WORK

The taxonomy proposed in [6] builds up a holistic point of view on the testing problem of intelligent vehicles. It is illustrated, how algorithms can be embedded on different platforms in simulations, which results in an increased realism, see Fig. 2.



(a) Simulation. (b) X-in-the-loop approach.

Fig. 2: X-in-the-loop: While the whole autonomous vehicle as well as the environment can be simulated, the algorithmic components-under-test are integrated on hardware or the vehicle. Source: [6].

In order to adapt simulations for perception-related tests, focus is set on the conceptional design of hardware-in-the-loop testrigs, e.g. for vision-based sensorics [7] or testing real-time vision-based lane detection and tracking [8]. A hardware-in-the-loop framework integrating SUMO has recently been presented in [9] with the focus on human, economic driving. Extending the hardware-in-the-loop concept leads to the vehicle-in-the-loop concept, integrating an experimental vehicle equipped with the algorithms-under-test. First fundamentals can be found in the VEHiL concept [10], which then are extended to proving ground with sufficient free space and inclusion of the driver using virtual and augmented reality techniques [11], [12].

Current research also focusses on the creation of flexible and modular frameworks for easily integrating ADAS components, such as the simulation framework *TRAFFIS* [13]. There are several commercial solutions for the development of advanced driver assistance systems (*ADAS*), but often focused on single aspects, such as sensor modeling or vehicle dynamics. Therefore, recent progress also has been presented with these commercial systems, like *SIVIC* [14], [15], [16] or *PreScan ITS Modeler* [17]. A more detailed overview on commercial solutions regarding advantages and disadvantages can be found in [18].

Also research in the field of robotics concerns with the development of frameworks, which facilitate the development of robotic skills, reaching from perception to acutation. For instance, *Gazebo* is already widely used in the robotics research community due to its modeling capabilities that incorporate mobile robots with a wide range of sensors, controllers and actuators [19], [20]. In [18] Gazebo has already been used as underlying framework for the design of an exemplary driving simulator, but a concrete integration of intelligent vehicles capabilities remains missing.

Another example for a robotics framework is *USARSim*, which is coupled with the traffic flow simulation Simulation of Urban Mobility (*SUMO*) developed by DLR [21] in [22]. The results lay the foundation for the present work. While their work focuses on the co-simulation of a robotics and a traffic simulator, we generalize this approach towards a generic, seamless development framework, also aiming from software- to vehicle-in-the-loop testing applications. We also revise the resulting development framework with respect to the definition and parametrization of test scenarios for high level components.

III. CONCEPT

In order to test and validate high level components for automated driving, such as trajectory planning or traffic participant prediction, traffic-based scenarios should be rather applied than fixed, standardized use-case catalogues, as we have already suggested in [6]. Case-by-Case evaluation, as for instance used for automated emergency braking assistance, can be conducted by defining traffic participants and maneuver chains manually. This is the typical approach by vehicle dynamics simulation tools. In order to achieve a high coverage of the situation space, traffic situations should be varied concerning the surrounding traffic flow.

If we consider the set of scenarios, which can be created within a specific simulation framework x_i, we can state out, that this is only a subset $\tilde{\Omega}_i$ of the infinite testing set Ω, see Fig. 3. Every simulation framework therefore covers a particular subset of generatable, varying environmental conditions.

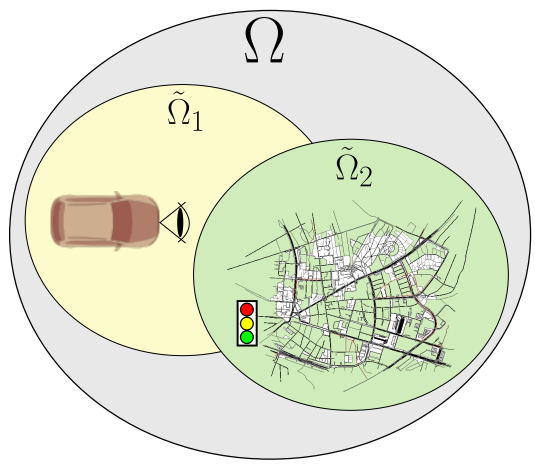


Fig. 3: The combination of simulation frameworks with different foci increases the coverage of the set of all traffic situations Ω.

By a smart combination of multiple single simulation instances, we increase the simulation capabilities of the resulting framework. This way the amount of possible simulated traffic scenarios comes closer to the optimum. We propose to modularize the generic task of simulation into subtasks. Each subtask is then fulfilled by a simulation, that is an expert on its specific domain. Although each simulation may only have reduced capabilities compared to an all-purpose simulation, we achieve a greater traffic scenario coverage than the latter.

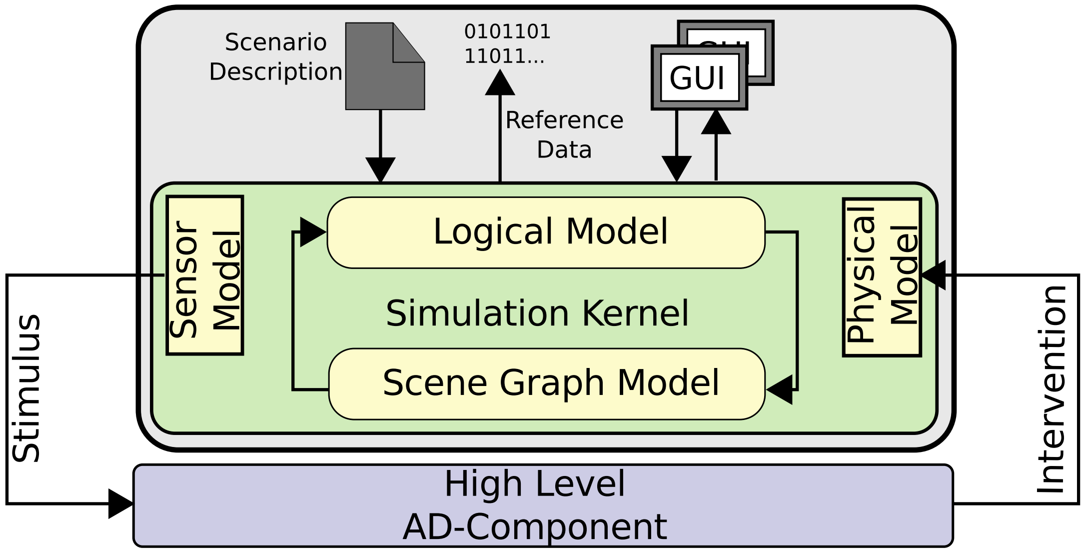


Fig. 4: Our proposed concept: The high level component-under-test is stimulated by the simulation, and an appropriate vehicle interface is called or - in the case of vehicle-in-the-loop - an avatar without any own mechanic element is moved according to the observed reaction of the experimental vehicle.

Also, it is necessary that a simulation framework provides interfaces in order to integrate algorithms at different scales of implementation, as well as integrating different algorithmic parts to evaluate their specific behavior, see Fig. 2. Therefore, appropriate stimuli have to be provided, such as simple sensory data from simulated sensor models up to higher-level information, for instance semantic description of the surrounding traffic participants' states in the simulated world model.

Therefore, we propose a general concept, see Fig. 4, which consists of a *Scene Graph Model*, a *Logical model*, as well as *Vehicle Mechanics* and a *Sensor Model*. The *Logical Model* handles the updating process of simulated traffic participants' states. The *Scene Graph Model* assigns geometric models and additional object properties to the traffic participants and static environment. The *Mechanics Model* provides capabilities to define kinematic and dynamic elements of a vehicle. The *Sensor Model* provides localization and perception models, which are used to derive restricted views on the virtual scene. On top of the kernel additional administrative components are dedicated to the definition of scenarios, such as picking reference data out of the simulation as well as user interfaces for interacting with the simulated traffic scenario. The framework's different parts are described in detail in the following sections:

A. Simulation Kernel

The *Logical Model* is dedicated to the creation of realistic road network and administration of realistic traffic flows. Road networks are based on geometrical and topological connections of roads and lanes, speed limitations and traffic regulations. These elements act as constraints for the resulting behavior of the traffic participants. It also handles the behavior update of the microscopic-defined[1] behavior of the traffic participants. The behavior is characterized considering an agent's profile with comfort parameters, like a maximum velocity or a minimum distance to the traffic participant driving ahead or observing adjacent lanes for lane-change maneuvers. If those agent behavior models assume a perfect perception, they can easily be updated without concerning perception models and therefore in a single step.

The *Mechanics Model* provides capabilities of defining geometric, kinematic and dynamic elements and properties of a vehicle. Thus, missing components of a vehicle can be simulated and provided to the high level component to control it. Depending on the necessary granularity, this can range from a simple description of the mass distribution and ackermann movement to a detailed description of single components like steering, suspension units and tires, finally interacting with the ground. By modeling suspension units and shock absorbers, a situation assessment component, which underlies a pitching sensor perception can be tested, for instance.

The *Scene Graph Model* represents an enriched environment model. It assigns visual models and geometric or semantic properties to the individual traffic participants as well as to static environment elements. Thereby, it administrates the link to the *Logical Model*, providing a compound information representation accessible for the sensor models. So, for instance, LIDAR-disturbing windows can be modeled setting an object's transparency.

The *Sensor Model* provides access to any kind perceivable information, such as localization or camera images or even abstract entity information, such as the semantic state of a

[1]The simulation of traffic participants generally can be distinguished between microscopic and macroscopic models. In contrast to macroscopic modeling, where traffic is defined in terms of thermodynamics, microscopic models are defined by artificial agents [23].

traffic light. In doing so a varying deviation from the ground truth information can be chosen. Thereby, sensor models are not restricted to pure sensory information, but also can be enhanced to a semantic level, which represents the environment on an entitiy level. Using physics-motivated simulation models, such as ray-casting to approximate pinhole cameras, the resulting entity information underlies occlusions or a restricted view due to the view frustrum. Also, they can be implemented based on other strategies, such as wavefront propagation for instance. This approach is reasonable for the testing and validation of high level components. Within the simulation kernel a loop takes place, defining a processing order and updating the internal states and broadcasting them to all kernel models leading to synchronization. An external loop connects the simulation kernel to the external high level component for automated driving, providing the desired input as stimulus. The component reacts and feeds its intervention back into the simulation kernel. Depending on the current application the internal and external processing loops may occur at different frequencies.

B. Embedding High Level Components

Open-loop systems, such as vehicle maneuver prediction, are not dependent on an external feed-back loop and can also be evaluated using logged data from test drives. We thereby focus on systems, which need to be tested *closed-loop*, which evaluate their feedback on the real world, for example on a simulated engine model. According to the high level component-under-test, the interfaces have to be defined and provided concerning two aspects: First, an appropriate interface with the means of control values, for instance the torque input of an engine, has to be provided. Second, in order to provide a seamless testing framework, algorithms have to be tested consecutively, for instance model-based (simulink), as compiled high level code (C++) or already compiled on an electronic control unit (*ECU*). So, the ECU then has to be coupled using a network or CAN interface with the appropriate inputs and outputs.

The *vehicle-in-the-loop* approach [11] represents a very deep integration of high level components. Thereby, the algorithms are integrated on an experimental vehicle driven on a spacious testing ground. The simulation kernel only receives the vehicle's pose observed by a high precision localization unit and the algorithms are stimulated by virtual sensory data. In this case, no vehicle mechanics simulation model is used, but replaced by a so called *avatar*. Here, a subset of simulation models is replaced by reality, especially concerning the ego-vehicle itself and the underlying road network with its properties, such as the friction coefficient. This causes additional requirements to be met: We have to assume, that the measured pose (GPS) as well as the modeled road network is coherent with the real world. This requirement can also be referred to as ground truth property [6]. The road network coherence can be achieved by either creating or adapting an existing map by D-GPS/IMU together with external sensor observations [24], for example.

C. Scenario Description

A simulation framework typically provides a domain specific language [25] for the definition of simulation scenarios. This description parametrizes the different kernel models in order to create the expected simulation scenario. In the present case, a common description for the parametrization of the kernel is necessary, covering the following aspects:

- The road network has to be defined with different constraints, such as traffic regulations or traffic lights, for instance.
- Static environment elements like house facades have to be defined.
- The amount of traffic participants and their behavior have to be parametrized, similary to the manner of civil engineers: using origin-destination matrices or distinct routes.
- Interfaces suitable for the connected high level component have to be defined.

D. Reference Views and Graphical User Interface

Reference views resemble the access points to reference data, which can be used in order to retrieve ground truth information about e.g. phyiscal quantities. Those can be used in order to reconstruct a system's behavior or evaluate its performance. Reference data can be transformed without information loss. Additional metrics can be calculated and provided for analyzing the performance of an algorithm.

User interfaces enable the interaction with the simulated traffic scenarios. The operator can interact with the different kernel modules, depending on what kind of manipulation, like starting or recording the testrun, or access of information he is interested in.

IV. REALIZATION

We've implemented the concept upon freely available frameworks such as the robotics simulation *GAZEBO*, the robotics middleware *ROS* and the traffic flow simulation *SUMO*, which is used as an exemplary instance of the logical model, introduced in Sec. III-A.

Our toolchain to create realistic traffic scenarios consists of the following steps: First, road networks are extracted from the OpenStreetMap format [26]. They are converted into a graph based road network description, which is readable by SUMO. Based on additional information in OpenStreetMaps, such as static environment, geometrical models can be exported to the *Scene Graph Model*. Based on the road network either random routes or fixed routes are created. This description is hold within a XML-based SUMO-internal format. Then, also the scenario description for the *Scene Graph Model* can be built up using the *Simulation Description Format* (SDF) [27], providing several elements for the definition of the static environment, the road network and the configuration of the high level component to the *Physical* and the *Sensor Model*. To instantiate the *Sensor Model*, the *Scene Graph Model* and the *Mechanical Model* we employ Gazebo.

A. Plugin System

Gazebo provides several kinds of plugins in order to map different skills to the freamework, see Fig. 5. It provides a high modularity, which makes Gazebo flexible and scalable and thereby attractive to employ models at different scales and with different interfaces: For instance, world plugins

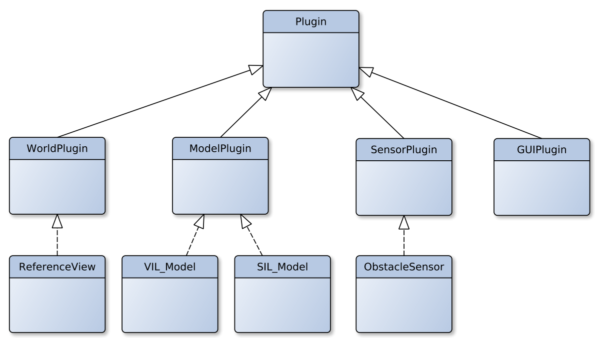


Fig. 5: The plugin interface is used for defining multiple models, sensors and view aspects in our simulation framework.

are suitable to create task-specific reference views. Entities can be accessed in the *Scene Graph Model* and converted to ROS topic messages, which then delivers ground truth information.

B. Embedding High Level Components

In order to evaluate our concept, we have realized two instances of vehicle representations with a varying degree of the underlying mechanics model. First, we have implemented a vehicle, which is controlable with throttle, brake and steering, see Fig. 6. Its level of detail also considers roll and pitch movements of the vehicle, by modeling suspension units at every wheel. Also the ackermann steering is considered for a physical plausible movement. Second, in order to evaluate the VIL-application, we have implemented an *avatar*, which does not contain any mechanics model, but also a geometric sensor configuration to receive virtual stimuli. This model receives its pose from the ROS transformation system. Thereby, any localization control unit simply have to provide a localization information between the world and the vehicle base frame. Both are built upon the unified robot description language (URDF) [28] an SDF [27], which facilitates the definition of kinematic chains based on XML-structures.

C. The Logical and Scene Graph Model

The environment model is linked with the running traffic flow. The interface is based on message types, which are sent over network. A message snapshot contains a representation of all dynamic element states, like vehicles and traffic lights. An additional structure checks every message if an appropriate representation in the *Scene Graph Model* is contained and adapts its properties.

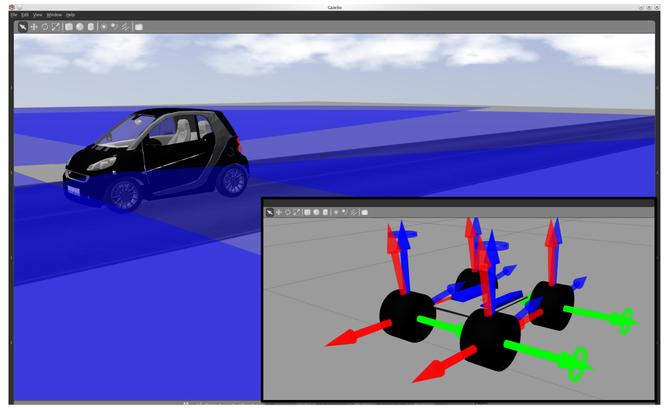


Fig. 6: Visualization of a detailed vehicle model: This model concerns aspects such as ackermann steering and suspension units, including actuators, controllers and sensors in order to develop autonomous driving functions. The blue sectors indicate the sensor ranges.

D. Sensor Model

In order to obtain abstract entity information for a high level component, we have implemented an abstract obstacle sensor. Casting rays in the *Scene Graph Model*, we simulate the view frustrum as well as occlusions, which could result from a camera, see Fig.7. This kind of abstract entity information is collected over every hit entity and provided as a list containing the entities' type (pedestrian, car, traffic light), its velocity or state. These lists then can be noised by an appropriate noise model and provided to the high level automated driving component, such as a trajectory planning algorithm.

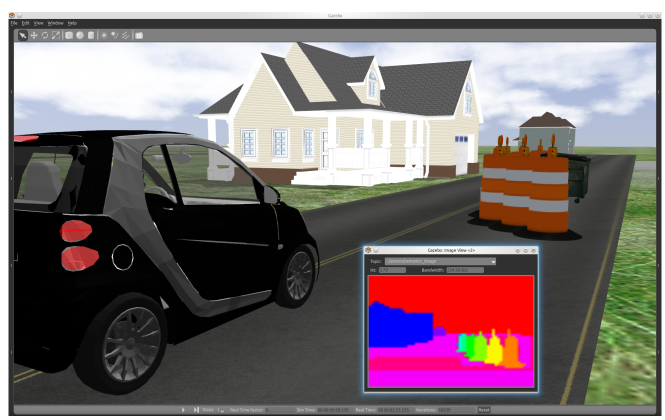


Fig. 7: Visualization of a virtual scene and the scene labels: A ray-casting algorithm is used to simulate a camera-based object classification. This abstract sensor then provides obstacle lists, containing information about the perceived obstacles.

V. CASE STUDY

In order to evaluate our concept, we use the vehicle-in-the-loop application with the FZI's experimental vehicle CoCar[2],

[2]For more information on CoCar, see `http://url.fzi.de/cocar`

which is equipped with a wide range of sensors and actuators for autonomous driving. To evaluate our concept, CoCar is driven on a real road network, with augmented, virtual surrounding traffic participants. Therefore, this road network has been modeled in SUMO and Gazebo, see Fig. 8. We assume, that the OpenStreetMap data is precise enough to be coherent with the real world and the derived road network bounded, so that the map can be adapted to reality by shifting the map with a constant offset. By approving the virtual ego positioning as well as the opposite traffic participants with sticking out landmarks like road bends, we could verify a plausible overlap between the real world and the derived road network.

An *avatar* model plugin containing the visual model and the abstract entity sensor configuration with CoCar is instantiated. The avatar's pose is updated in the simulation kernel by a high precision localization unit based on D-GPS/INS, which provides an estimation with an accuracy of ± 2cm.

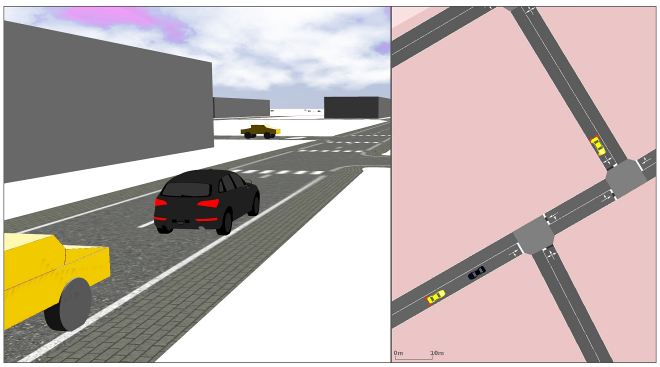


Fig. 8: The modeled testing ground has been equipped with virtual traffic participants from the traffic flow simulation SUMO: The black *avatar* is moved according to the measured movement of the real experimental vehicle, whereas the yellow vehicles' behaviour is determined by SUMO.

Our experiments have shown that the runtime depends on the degree of detail used in the visual scene representation. On a computer system with common hardware, the framework's processing loop is real-time capable when a slightly decreased detail level for the traffic participants' visual models is used.

In our tests, we were able to produce realistically driving traffic participants on real road networks that were augmented into our ego vehicle's environment. We could also vary the partial observability of the virtual environment, modifying the amount of traffic participants perceived by the ego vehicle. Thus, realistic simulation of a changing sensor horizon was possible. In any case, the ground truth data could be accessed in order to compare measurements and analyze unexpected performance of automated driving functions.

VI. CONCLUSION

Our concept demonstrates the application of a co-simulation between a *Scene Graph Model* including mechanics and sensor modeling and a traffic flow simulation acting as an *Logical Model* in order to test and validate high level automated driving components, such as trajectory planning or vehicle prediction. The focus was on showing the application of such a co-simulation in different integration levels. Instead of defining traffic participants with maneuver chains, this approach results in the definition of traffic scenarios by parametrization of the number of agents and the underlying models' parametrization, which explicitely increases the coverage of the testing space Ω. This represents an approach away from the fixed use-case catalogues towards traffic-based evaluation. Future work will focus on the question, if by the integration of an additional dimension, namely the behavior model space itself, Ω can be approximated further and therefore, results in additional traffic scenarios.

Instead of focusing on modeling physical sensor models, which would require photo-realistic rendering environments for vision-based sensors e.g., we propose to use higher simulation models, which are inspired by physical perception principles. Therefore, the *Scene Graph Model* lays the foundation for an enhanced environment model, which facilitates the inclusion of multiple properties of entities and to generalize sensor data in order to provide high level information, for the stimulation of high level components.

We also have shown how our testing framework targets different integration levels. A specific sight has been put into the vehicle-in-the-loop application. In order to create realistic scenarios, the validity of the underlying road network, here OpenStreetMap, has to be assured and a precise localization unit, which is decoupled from the system-under-test and the testing loop, has to be used. Compared to real test drives, this provides the advantage that realistic road networks can be filled with virtual traffic participants in order to evaluate the performance on real vehicles enabling safe testing.

Our framework will be extended with the focus on the following drawbacks we have encountered during this work: We will focus on the questions, how the ground truth property for the map can be automatically ensured with OpenStreetMap data. There are several practical problems, resulting mainly from the selected traffic flow simulation. At the very moment, only 2D road networks can be modeled. Therefore, scenarios such as highway entrance are hardly creatable. Traffic participant behaviour is mainly restricted to longitudinal behavior and teleportation instead of smooth lane changes occur. This is a very strong restriction for testing lane change prediction of highway pilots, for instance. Also the *Scene Graph Model* has to be extended in order to map relations and interactions between different entities, such as pedestrians entering a bus. Also we are interested in, enhanced the concept by including additional logical models, for instance, modeling weather aspects.

ACKNOWLEDGMENT

This work was partially supported by the project LogDaSim under the Federal Ministry of Education and Research (BMBF) grant 01IS14024C.

REFERENCES

[1] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn *et al.*, "Making bertha drive—an autonomous journey on a historic route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014.

[2] R. Kohlhaas, T. Schamm, D. Lenk, and J. M. Zöllner, "Towards driving autonomously: Autonomous cruise control in urban environments," in *Intelligent Vehicles Symposium Workshops (IV Workshops), 2013 IEEE*, June 2013, pp. 109–114.

[3] Alberto Broggi, Pietro Cerri, Stefano Debattisti, Maria Chiara Laghi, Paolo Medici, Matteo Panciroli, Antonio Prioletti, "Proud - public road urban driverless test: Architecture and results."

[4] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, June 2011, pp. 163–168.

[5] ISO, "Road vehicles – Functional safety," 2011.

[6] J. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels, and J. M. Zöllner, "Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, Sept 2015, pp. 1455–1462.

[7] F. Schmidt and E. Sax, "Funktionaler softwaretest für aktive fahrerassistenzsysteme mittels parametrierter szenario-simulation," in *Informatik 2009*, ser. LNI, S. Fischer, E. Maehle, and R. Reischuk, Eds., vol. 154. Bonn, Germany: GI, 2009.

[8] F. Coskun, O. Tuner, M. Karsligil, and L. Guvenc, "Real time lane detection and tracking system evaluated in a hardware-in-the-loop simulator," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, Sept 2010, pp. 1336–1343.

[9] W. Griggs, R. Ordonez-Hurtado, E. Crisostomi, F. Hausler, K. Massow, and R. Shorten, "A large-scale sumo-based emulation platform," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 16, no. 6, pp. 3050–3059, Dec 2015.

[10] O. Gietelink, J. Ploeg, B. De Schutter, and M. Verhaegen, "Development of a driver information and warning system with vehicle hardware-in-the-loop simulations," *Mechatronics,* Special Issue on Hardware-in-the-Loop Simulation, vol. 19, no. 7, pp. 1091–1104, Oct. 2009.

[11] T. Bock, M. Maurer, and G. Farber, "Validation of the vehicle in the loop (vil); a milestone for the simulation of driver assistance systems," in *2007 IEEE Intelligent Vehicles Symposium*, 2007, pp. 612–617.

[12] S. Schwab, T. Leichsenring, M. R. Zofka, and T. Bär, "Consistent test method for assistance systems," *ATZ worldwide*, vol. 116, no. 9, pp. 38–43, 2014.

[13] K. Abdelgawad, B. Hassan, M. Grafe, and I. Gräßler, "A modular architecture of a PC-based driving simulator for advanced driver assistance systems development," in *Proceedings of the IEEE 15th International Workshop on Research and Education in Mechatronics REM*, 2014.

[14] A. Belbachir, J.-C. Smal, J.-M. Blosseville, and D. Gruyer, "Simulation-driven validation of advanced driving-assistance systems," *Procedia - Social and Behavioral Sciences*, vol. 48, no. 0, pp. 1205–1214, 2012, transport Research Arena 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877042812028327

[15] D. Gruyer, M. Grapinet, P. De Souza, "Modeling and validation of a new generic virtual optical sensor for adas prototyping," *Proceedings of the Intelligent Vehicles Symposium*, 2012.

[16] D. Gruyer, S. Choi, C. Boussard, and B. d'Andrea Novel, "From virtual to reality, how to prototype, test and evaluate new adas: Application to automatic car parking," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, 2014, pp. 261–267.

[17] M. Tideman and M. van Noort, "A simulation tool suite for developing connected vehicle systems," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, June 2013, pp. 713–718.

[18] K. Swanson, A. Brown, S. Brennan, and C. LaJambe, "Extending driving simulator capabilities toward hardware-in-the-loop testbeds and remote vehicle interfaces," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, 2013, pp. 122–127.

[19] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, Sep 2004, pp. 2149–2154.

[20] C. E. Aguero, N. Koenig, I. Chen, H. Boyer, S. Peters, J. Hsu, B. Gerkey, S. Paepcke, J. L. Rivero, J. Manzo *et al.*, "Inside the virtual robotics challenge: Simulating real-time robotic disaster response," *Automation Science and Engineering, IEEE Transactions on*, vol. 12, no. 2, pp. 494–506, 2015.

[21] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, December 2012.

[22] J. L. Pereira and R. J. Rossetti, "An integrated architecture for autonomous vehicles simulation," in *Proceedings of the 27th annual ACM symposium on applied computing*. ACM, 2012, pp. 286–292.

[23] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Pearson Education, 2003.

[24] M. R. Zofka, F. Kuhnt, R. Kohlhaas, C. Rist, T. Schamm, and J. M. Zöllner, "Data-driven simulation and parametrization of traffic scenarios for the development of advanced driver assistance systems," in *Information Fusion (Fusion), 2015 18th International Conference on*, July 2015, pp. 1422–1428.

[25] C. Berger, "Design considerations for a cyber-physical testing language on the example of autonomous driving," in *Proceedings of the 2012 Workshop on Domain-specific Modeling*, 2012, pp. 49–54.

[26] M. M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, Oct. 2008. [Online]. Available: http://dx.doi.org/10.1109/MPRV.2008.80

[27] Sdf-simulation description format. [Online]. Available: http://sdformat.org/

[28] Urdf-unified robot description format. [Online]. Available: http://wiki.ros.org/urdf