Collaborative Development of a Test Environment for Automated Driving

Dr. Tim Fricke, Dr. Falko Schuck, Dr. Jens Vogelpohl, BMW Group, Munich; Dr. Johannes Ax, Dr. Ralf Stolpe, Dr. Hendrik Amelunxen, dSPACE GmbH, Paderborn

Abstract
Delivering ever more automated driving (AD) functions presents the entire automotive industry with various challenges. High-performance computers (HPCs) are required to handle the complex environment perception and decision making processes and they must be integrated with various environment sensors. As the level of automation increases – and with it the complexity of the vehicle and its functions – more and more test kilometers must be driven and more and more test cases must be passed. In a test strategy that emphasizes virtual testing for the sake of efficiency, hardware-in-the-loop (HIL) testing of HPCs and of the combined system of HPC and sensors plays a key role, bridging the gap between prototype and fully virtual testing (software- or model-in-the-loop, SIL/MIL) and enabling system integration tests. Although HIL testing is a well-established test method that is used for more than 20 years in the automotive industry, its application for HPC and system-level testing of AD systems brings new challenges with it, on the technical level as well as on the organizational level.

The main technical challenge is the highly complex simulation of the vehicle environment, physical sensor characteristics and all relevant data bus communication with a higher maturity and robustness than the system under test. Beyond that, a certain technical continuity between test platforms (prototypes, HIL, SIL) must be ensured to enable cross-validation of methodologies and to increase efficiency. From these technical challenges arises the need to bring together various expert parties (OEM, suppliers, test system providers, engineering contractors) in an effective and efficient collaboration.

This paper presents how these challenges have been met in the scope of an OEM’s development for series production. It describes the overall test strategy and how different simulation and testing approaches were applied to test AD functions. Special focus is drawn to HIL testing of HPCs, for which the technical characteristics of the test system are outlined.

On the organizational level, joint development of test technologies, consequent use of technical standards such as FMI and ASAM Open-X, early pre-integration of test system
components, joint feature backlogs and overarching project organizations are presented as key success factors.

**Introduction: Challenges of Developing and Testing Automated Driving Systems**

Over the last 30 years, automotive electronics and software have mainly been used to develop innovations that made driving and using vehicles safer, more efficient and more comfortable. Most of the functionality that has been brought into vehicles was using a set of – relatively – simple sensors, based on basic physical principles. With the introduction of advanced driver assistance systems and the goal of highly automated driving, this changes dramatically. Vehicles interact more and more with their surroundings, and more and more tasks, such as steering, braking and driving, are to be automated. This leads to an immense increase of automation and safety functions, which often have to be fail-operational to allow the driver to take over or to bring the vehicle to a safe halt [1]. In order to achieve this, AD systems must be able to obtain a highly comprehensive, precise and reliable understanding of the vehicle surroundings. Therefore, AD systems utilize a lot of sensors, such as cameras, radars, lidars and ultrasound sensors. By combining all this sensor information, the automated vehicle creates its own perception of its environment. Based on this, it can plan and act accordingly.

While former vehicle functions mainly have been control systems with quite a limited amount of sensor data, AD systems require an immense amount of data, not only for driving, but also for machine learning and for validation [2]. Also, the number of different scenarios that are required for machine learning and validation, cannot be covered by real test drives any more. The extensive use of simulation becomes a critical factor for the validation of AD systems. Therefore, classical validation methods, such as prototype and HIL testing (closed-loop) must be complemented by other methods [3]. SIL testing, as an example, allows for large scale, parallelized simulation and scenario-based validation. Software (SW) and hardware (HW) reprocessing on the other hand utilize real-world data, recorded in lots of vehicles, and allow to re-run these recorded real-world situations open-loop for test purposes.

Eventually, for validating AD systems, a safety-based test strategy is required, that combines these test methods for the different development phases and system levels in an intelligent, effective and efficient way (see Fig. 1).
In this paper, we focus on HIL testing of AD-HPCs using closed-loop real-time HIL testing. Developing and operating HIL test systems for AD-HPCs poses a lot of challenges:

- First, a comprehensive simulation of the vehicle’s environment perception is required. This includes the simulation of all the sensors that are equipped in the real vehicle, the generation of their sensor data, and feeding this sensor data into the control system, the AD-HPC. Communication over buses and networks must also be provided and simulated, including the possibility to inject failures in order to test the fail-operational behaviour. All of this has to happen in real time. In setting up such a HIL test environment, many different tasks and collaborators need to be coordinated. This includes the OEM (responsible for the validation and homologation of the AD system), sensor suppliers (providing specifications and models for their sensor technology), the test system provider, plus a couple of engineering contractors (for different tasks, such as model integration and delivery, test automation or system maintenance).

- Since HIL testing is only one element of the test strategy, it must work seamlessly together with other test methods, such as SIL and SW/HW reprocessing. This can pose requirements on the process level, for example to put all test systems and methods under one central test management. Furthermore, synergies are important between the different test environments, for example a seamless re-use of simulation models between SIL and HIL, or a re-use of the same restbus configurations between HIL and HW reprocessing.

- An AD-HPC has to satisfy a multitude of use-cases and requirements. Moreover, software development today continues beyond a vehicle’s start of production (SOP), to deliver up-to-date user experiences via remote software updates. As a result, incremental
development must be employed not only for the AD-HPC software but also for the test system software. An agile development approach with highly frequent deliveries that need to be tested imposes additional challenges on test systems.

System Architecture for Hardware-in-the-Loop Testing of AD Systems

The system architecture of a HIL test bench for AD-HPC testing is mainly governed by the test strategy, the actual test cases and the HPC interface requirements, but it also plays a key role in facilitating an effective collaboration between all parties involved. With that in mind, the following system architecture has been chosen.

![Fig. 2: Schematic of the HIL system architecture](image)

Fig. 2 gives an overview of the HIL system and its components and their responsible party. The central component of the HIL test bench is a real-time computer (RT-PC), which provides the communication interface simulation for the system under test. Besides CAN and FlexRay
interfaces, Ethernet (SOME/IP) plays a major role here. The clock of the RT-PC is synchronized with that of the HPC, so that simulated messages can be correctly time-stamped.

Apart from the communication interfaces, the HPC also receives raw sensor data streams (camera, lidar) that are injected through a sensor interface unit (SI-Unit). The SI-Unit establishes the sensor specific electrical connections (e.g. GMSL, FPD-Link, CSI). Image generation for camera and lidar simulation is performed on a separate computer (Rendering-PC). Different layers of sensor models are integrated on the Rendering-PC (GPU based) and SI-Unit (FPGA based), simulating parts of the sensor physics. The sensor models can be divided in vendor specific sensor models (SM1) and generic sensor models (SM2). Very time critical parts of these specific sensor models are executed on the SI-Unit to allow a fast feedback/control channel of the AD-HPC to the sensor simulation. Other parts of the sensor models are integrated on the Rendering-PC. For each sensor layer the expertise of the different parties is required (see section Organizational Set-up and Facilitators). For image or lidar point cloud generation the Rendering-PC receives relevant scenery and motion data from the RT-PC.

The RT-PC indeed simulates the vehicle dynamics and the vehicle surroundings (static road and scenery, dynamic traffic). It also runs simulation models of other ECUs, such as the braking and steering systems, or provides static restbus simulation wherever sufficient. Simulation models of sensor-ECUs such as radar, ultrasound or satellite positioning sensors, run in a dedicated model environment on a separate computer (ME-PC). These simulation models are different from those on the Rendering-PC and the SI-Unit in that they provide the same Ethernet SOME/IP communication interfaces as their real ECU counterpart. They are therefore not mere sensor models, but combined sensor/logic models (SM/LM), comprising the simulation of both sensor physics and ECU behavior. Their output can be manipulated in the RT-PC, which enables the correction of time-stamps and the deliberate injection of errors. For some test system set-ups, it is possible to replace camera and lidar simulation on the Rendering-PC with corresponding sensor models on the model environment computer (ME-PC).

Finally, a human operator or a test automation system can manage the test bench from a host-PC. Test bench management includes the set-up of the simulated vehicle surroundings. Since the AD-HPC compares its perceived environment with maps of the real world, the simulated
environment must conform to parts of the real world. A toolchain of map converters and importers has therefore been set up to automatically generate a simulation of real roads.

The modular architecture allows partial set-ups of the test bench for early integration and testing of its components. This constitutes an important success factor. Considering that not only the AD-HPC itself, but also the test system around it is highly complex, a big bang integration would be a very arduous task that may not be finished early enough to provide testing capabilities on time. With a modular test system architecture and timely pre-integration activities, on the other hand, test system maturity is adequate for testing much earlier. This modular approach, however, also requires a tight specification of interfaces to prevent communication mismatch. The following interface standards can therefore be considered critical components of the system architecture.

ASAM OSI [4] plays a crucial role as interface between RT-PC and ME-PC, as well as between the sensor and logic models inside the ME-PC. The simulated vehicle environment (ground truth) with information about road, scenery, dynamic and static objects is provided by the RT-PC as OSI::SensorView. The sensor models on the ME-PC take OSI::SensorView as input and provide perception results as OSI::SensorData. The calculation of the ground truth data within the sensor area around the ego vehicle and its provision to the sensor models on the ME-PC and to models on the RT-PC is achieved within short sample times under hard real time conditions, incorporating Google Protocol Buffers [5] (as mandatory for ASAM OSI).

The sensor/logic model outputs (ME-PC) as well as the RT-PC’s interfaces with the HPC are compliant with the OEM’s message catalogue and the relevant AUTOSAR specifications [6]. Moreover, the sensor/logic models are Functional Mock-ups (FMUs) according to the Functional Mock-up Interface (FMI) standard [7]. The map conversion toolchain makes use of the Navigation Data Standard (NDS) [8] and ASAM OpenDRIVE [9] as exchange formats.

Organizational Set-up and Facilitators
A suitable system architecture alone is no guarantee for a successful project. Especially in the case of AD-HPC testing, where a multitude of expert parties need to bring in their know-how, an effective organizational set-up is required.

Being responsible for the final software and system release into production (and thereby to the end customer), the OEM of course takes charge of the overall coordination, specifying test
strategy and test systems. Test systems like the exemplary AD-HPC HIL are made up of many components from various suppliers, contractors, and the OEM itself. One might therefore imagine a centralized approach in which each component is specified and delivered separately, with the OEM acting as the single and central test system integrator.

A key success factor in the project under discussion, however, was the set-up of a more decentralized collaboration. Implementation, integration, verification, and validation activities were intelligently distributed across the network of collaborators, while the OEM of course retained control over the final system specifications and capabilities.

In order to meet the system requirements for all sensor data interfaces of the AD-HPC, a strong collaboration between all parties is required during the development process. For sensor raw data injection (via the Rendering-PC, [3]), different layers of sensor models are required with some parts strongly depending on the supplier-specific sensor types. Therefore, the test system provider (TSP) must work together with the tier 1 and tier 2 suppliers to develop sensor models which are executed on different components of the HIL test bench (see Fig. 2). The TSP delivers generic sensor models on the Rendering-PC which can be configured and parametrized by the tier 1 supplier. In addition, a post-processing interface is provided to manipulate the generated data, e.g. by integrating the tier 1 supplier's own sensor models before the raw sensor data is transmitted to the SI-Unit. For the vendor specific sensor models the tier 2 suppliers provide datasheets and developer guides of the different sensor parts (e.g. imager) to the tier 1 supplier and the TSP. The electrical interfaces for the specific sensor types, defined by tier 1 supplier and OEM, are provided by the SI-Unit, which is a modular system with general support for all sensor interfaces available on the market. Development and pre-integration of the raw data injection can be done using a partial test system setup, leaving out for example the ME-PC and parts of the restbus simulation.

The sensor/logic models on the ME-PC are developed by the tier 1 suppliers according to the OEM's requirements. To facilitate this process, the ME-PC is set up stand-alone by the tier 1 supplier. It is a commercial-off-the-shelf computer with a model environment provided by the OEM. The OEM also provides templates for the sensor/logic models that ensure seamless interfacing with the rest of the test system and traces from the environment simulation that the tier 1 supplier can use for testing. For highly realistic simulation, it is possible to re-host the actual ECU code inside the sensor/logic model.
To enable an agile development process as well as continuous integration and testing, a triangular relationship between OEM, tier 1 supplier and test system provider (TSP) has been set up (Fig. 3). Therefore, a common definition of firmware versions is necessary to enable a functional interaction of all system components. With provisions from the OEM (ME-PC, AUTOSAR databases) and the tier 1 supplier (ECUs, information) the TSP can develop and test his products continuously already against the most realistic customer configurations in order to test the OEM-specific features. In addition, the tier 1 supplier operates identical test benches as the OEM to allow for an optimal knowledge exchange between all parties. The TSP ensures the operational capability of the tier 1 supplier directly after being nominated by the OEM. Organizational acceleration mechanisms in this context are the provision of test licenses and test hardware without having to wait for long purchasing processes.

The ASAM Open-X standards \[4,9,12, \ldots\] play a crucial role in the system architecture and the validation process. Especially ASAM OSI, which serves as link between modules from the OEM, the TSP and tier 1 suppliers, is of central importance. Since ASAM OSI is a rather young standard which also has overlaps in content with ASAM OpenDRIVE, it has been proved as
an important and promising part, to actively take part in the ASAM OSI development project to further clarify and harmonize the standard to make it more robust and mature.

**Collaborative Development of Test Systems for AD-HPCs**

To tackle the many challenges of developing test systems for AD-HPCs, a well-suited system architecture and a supporting organizational set-up have been identified as important success factors in the previous sections. The third key ingredient for success, bringing life to the technical and organizational framework, is the collaborative development of the test system by OEM, tier 1 suppliers and TSP.

In the project under discussion, an agile cooperation has been set up that is characterized by a (bi-)weekly harmonization and prioritization of test system requirements between the project partners and the implementation of these requirements in the scope of recurring “product increments”. All issues were collected and managed in a common backlog, which also served as a common knowledge base. New project partners, like tier 1 suppliers that have newly been nominated, could thus be integrated swiftly. Access to the accumulated know-how and the current status of test system development was granted within few days after entry into the project, enabling effective collaboration right from the start. Lengthy procurement processes were bridged by the provision of intermediary hard- or software (e.g., evaluation licenses) to enable early familiarization.

To maximize the contribution of a test system to product development, it must be available earlier and at a higher degree of maturity than the system under test itself. Moreover, test system requirements evolve as the system under test matures. Therefore, areas where extensive test system qualifications will be necessary (e.g., a new AUTOSAR version, new communication protocol peculiarities, new hardware interfaces) must be identified and addressed with proof of concept studies early in the project. Newly implemented test system capabilities should then be validated and verified on a prototype test bench, which comprises a simulation of the system under test.

We deem face-to-face collaboration extremely important for the success of a project like the one under discussion. Unfortunately, though, all project partners were forced to stay at home for most of the time in recent months due to the COVID-19 pandemic. Remote commissioning and operation of complex test systems for AD-HPCs with up to 6 computer screens, a multitude of LEDs on hardware components and potentially the smell of overvoltage, is simply not
efficient. Instead, experts from each party should be on site, jointly developing, commissioning and operating the test systems. As a side effect, test system providers become intimately familiar with the OEM’s and the tier 1 suppliers’ day-to-day operations and can continuously adapt their products to their customers’ needs.

Changes to the test system are sometimes required urgently and on short notice. The joint coordination of all parties in the agile development process presented allows a quick reaction, taking into consideration possible workarounds. This requires a high degree of mutual transparency and trust, which is favored by individual cooperation and involvement. Coordination meetings must also regularly be joined by management to accelerate large-scale decision processes.

All in all, the collaborative development method enabled the timely implementation of many test system requirements, the early identification and planning of new and future requirements and quick reactions to requested changes in test system behavior. It has been proven in the introduction of new technologies, like automated driving, and the associated test strategies. The open, intensive, regular collaboration between project partners emphasizes individual cooperation and requires OEMs and suppliers alike to adapt.

Conclusions and Outlook
Automated driving requires the use of cutting-edge technology. Therefore, it is not surprising that any test environment for automated driving, such as an AD-HPC HIL test bench, needs to employ state-of-the-art test system components. Huge amounts of data must be generated, manipulated, synchronized and finally transmitted to the HPC. To tackle this challenge in an effective and efficient manner, we identified the following success factors:

First, a modular test system architecture was designed, so that parts of the test system could be set up on-site at each collaborator. This allowed pre-integration of test system components, which in turn largely facilitated final test system integration at the OEM. As a result, testing capabilities could be provided earlier. Important enablers for our modular test system architecture were interface standards such as FMI, ASAM OSI, AUTOSAR and ASAM OpenDRIVE.

Second, test system development tasks were distributed across OEM, tier 1 (and tier 2) suppliers and test system provider. Hence, the expert know-how of each collaborator was fully
utilized, yielding a test system of high maturity. Most notably, the tier 1 and tier 2 suppliers brought in all their detailed knowledge of sensor physics and sensor-ECU behavior. The test system provider directly collaborated with the tier 1 and 2 suppliers to enable raw data injection based on the simulated environment. The OEM managed the requirements for the overall test system and all its components, ensured interface conformity (especially with AUTOSAR and ASAM OSI) and facilitated sensor/logic model development by the tier 1 supplier with the provision of its model environment.

Third, the set-up of an agile day-to-day collaboration between OEM, tier 1 suppliers and test system provider considerably accelerated the pace of development. Important elements of our collaboration were shared backlogs and knowledge bases, highly frequent meetings for both technical discussions and management decisions, and regular on-site presence of all parties involved. Of course, modern issue tracking and agile project management tools were used throughout the collaboration.

Managing complexity by modularization and pre-integration in a collaborative development effort will become more and more important as the level of automation in vehicles will increase. We therefore deem a continuation of the presented strategy in future projects extremely valuable. Beyond that, early-prototype test benches should be employed even more extensively. We expect that this would further accelerate the test system development. Moreover, all interface standards mentioned should be continuously refined and advanced, because they are crucial for an efficient multilateral collaboration. In the future, however, not only interface standards will be shared across the industry. Like already the case for mapping data of the real world [10], maps and test scenarios in the virtual world will be distributed across OEMs [11]. For scenario-based testing, this is facilitated by the ASAM OpenSCENARIO interface standard [12], which should also be supported by all relevant test system tools. All in all, the consequent continuation and expansion of the collaborative development strategy presented will help bringing higher levels of automated driving to a broad range of customers.
References


[9] ASAM OpenDRIVE homepage: https://www.asam.net/standards/detail/opendrive/


[12] ASAM OpenSCENARIO homepage: https://www.asam.net/standards/detail/openscenario/