Deliverable D12

Test Orchestration and Test Execution – Process, Methods and Example Realizations

Version 1.0

Editor Martin Dörr (ZF) and Stefan Otten (FZI)

Projektkoordination Robert Bosch GmbH und BMW AG

Fälligkeit 12.12.2023

Erstellungsdatum 13.12.2023

Veröffentlichung 29.12.2023

Freigabe INT, LI, PU

INT: Freigabe projektintern, Partner, wissenschaftliche Unterauftragnehmer und Projektträger (betrifft die Anwendungen von Konzepten)

LI: Freigabe VDA-LI, andere Projekte der Projektfamilien und Mitglieder der VDA-Leitinitiative (betrifft konzeptionelle Ergebnisse)

PU: Freigabe für die Öffentlichkeit

Gefördert durch:

Bundesministerium für Wirtschaft und Energie

aufgrund eines Beschlusses des Deutschen Bundestages

Dokumenteninformation

Autoren

M. Dörr (ZF), J. Pfisterer (Valeo), C. King (FZI), B. Filzek (Continental), T. Menzel (TUBS), J. Fuchs (FZI), A. Spot (dSPACE), H. Hungar (DLR), G. Temme (DLR), G. Schunk (Valeo), M. Zipfl (FZI), H. Padusinski (FZI), J. Millitzer (FhI LBF), W. Ritter (Mercedes), D. Rottler (Mercedes), Thomas Corell (Continental), S. Krebs-Radic (ZF), E. Fox (Akka), S. Otten (FZI)

Review durch

TP7

Kontakt

Martin Dörr	Stefan Otten
ZF Friedrichshafen AG	FZI Forschungszentrum Informatik
88046 Friedrichshafen	76131 Karlsruhe
Deutschland	Deutschland
Phone: +49 7541 77 6691	+49 721 9654 156
Email: Martin.Doerr@zf.com	Otten@fzi.de

Revisionslog

Version	Datum	Kommentar	Autor	Partner
0.1	14.11.2023	Initial document	Martin Dörr	ZF
0.2	23.11.2023	Chapter 3	Jonas Pfisterer	Valeo
0.3	28.11.2023	Chapter 2	Martin Dörr	ZF
0.4	28.11.2023	Chapter 3	Hubert Padusinski	FZI
0.5	29.11.2023	Chapter 3	Jonathan Millitzer	FhG LBF
0.6	29.11.2023	Chapter 3	Erik Fox	Akka / BMW
0.7	29.11.2023	Chapter 3	Maximilian Zipfl	FZI
0.8	30.11.2023	Chapter 3	Alexander Spot	dSpace
0.9	01.12.2023	Chapter 3	Daniel Rottler	Mercedes
0.10	01.12.2023	Chapter 3	Thomas Corell	Continental
0.11	04.12.2023	Chapter 3	Gerald Temme	DLR
0.12	06.12.2023	Chapter 3	Slavisa Krebs-Radic	ZF
0.13	08.12.2023	Chapter 3	Roland Galbas	Bosch
0.14	11.12.2023	Chapter 3	Stefan Otten	FZI
0.15	12.12.2023	Chapter 1&4	Martin Dörr	ZF
1.0	13.12.2023	Final document edited	B. Filzek, S. Otten, M. Dörr	Conti, FZI, ZF

Table of content

Abbildungsverzeichnis

1 Introduction

This deliverable D12 represents the result of VVM's sub project TP7 on test procedure. On the one hand side, it describes the methodological approach of test orchestration and on the other hand side, it presents example test implementations.

The concept of Test Orchestration with its processes and methods has been developed via the deliverables D4 (Initial Concept of Test Orchestration) and D7 (Final Methodology of Test Orchestration). Chapter 2 describes the final state of the concept of Test Orchestration.

Chapter 3 focuses on the application of Test Orchestration to enable seamless testing. It provides multiple examples, that have been realized by the partners of VVM and that show different approaches to realize test infrastructures.

This document is based on the glossary developed in subproject 1.

2 Test Orchestration - Test Execution - Test Evaluation

2.1 Localisation within the Assurance Framework

The Assurance Framework provides a representation of the processes that are necessary from conception to the release of a technical system. As part of VVM, it answers the question of what needs to be done to develop an ADS Level 4/5 ready for the market. The "why" is implicitly included. Beyond the individual process steps, however, the overview also includes their essential links reflecting the logical dependence of single steps in terms of content. For the sake of clarity, these dependencies are only shown forwards in the course, i.e. iteration loops and feedbacks at the interfaces are of course present, but not separately illustrated.

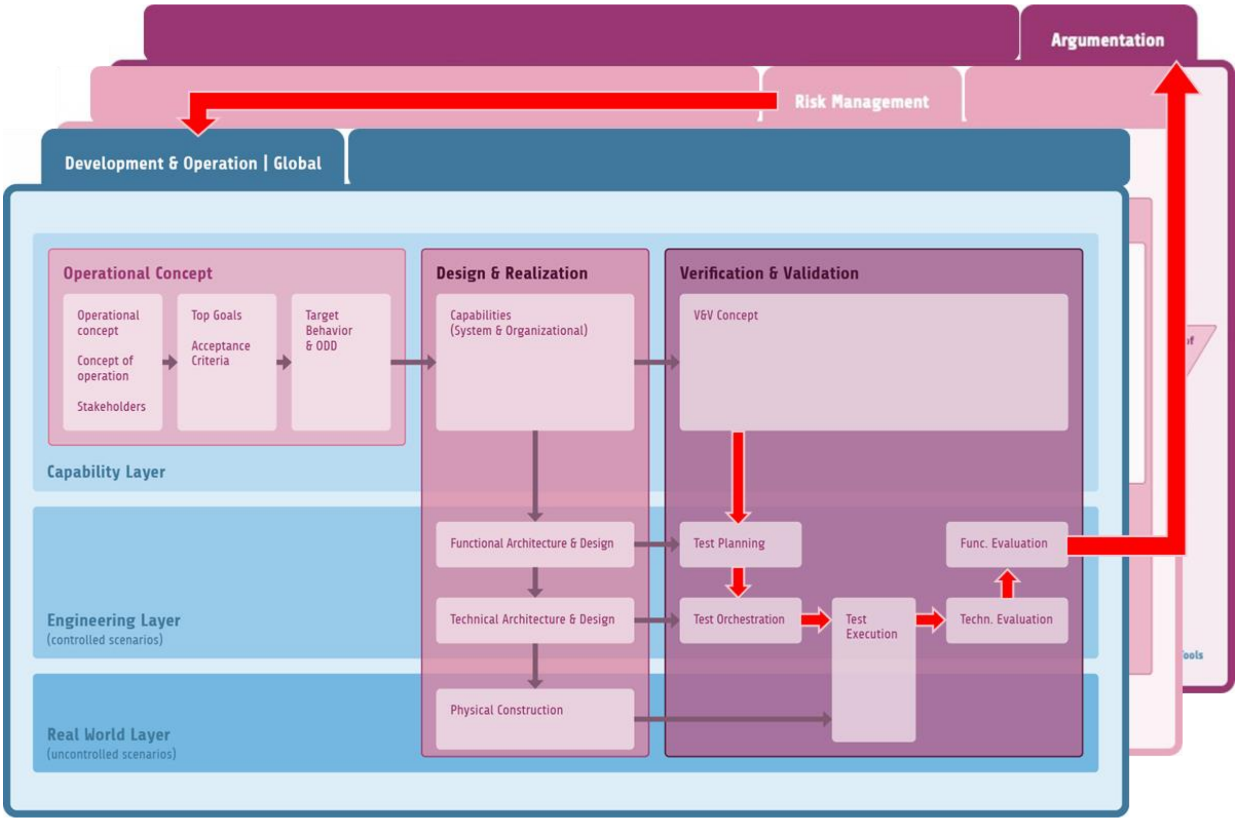


Figure 1: V&V within the Assurance Framework

The global perspective of Development & Operation of the Assurance Framework shows the Test Orchestration within the V&V part. Important links exist to the perspectives of Risk Management and Argumentation. The Risk Management provides the risk model as prerequisite for the design process and for the aggregation of results within the functional evaluation. The Argumentation is the addressee of the evaluation results.

According to the concept of dividing into functional and technical level, Test Orchestration follows Test Planning and is dependent from the technical design.

2.2 Test Orchestration

Test orchestration must establish the link between test planning on the one hand and test execution on the other. It thus helps to ensure that the safety argumentation can be fully conducted in the implementation of the V&V concept. In doing so, it must always keep the issues of efficient testing, feasibility, significance, and coverage of the test space in mind.

More concrete, test orchestration is about translating the functional test specifications into technical test specifications. The main difference is the transition from the consideration of solution-neutral functions to the consideration of the real world of implemented and existing test objects and instances. Analogous to the functional test specification, a technical test specification contains all the detailed test requirements that are needed in this case for the actual execution of a single test case or a group of similar test cases. The main aspects are the definition of the test object (e.g., components, assemblies, subsystems or the overall system), the selection of the test instance (e.g., SiL, HiL, test bench, proving ground or field) including the description of the test scenario and the description of the test objectives, which contain the specific measured variables to be examined.

2.2.1 Inputs into Test Orchestration

2.2.1.1 Functional Test Specification

The degree of freedom to determine test object, test instance, test scenario and test objectives varies for verification and validation. Thus, the functional test specifications are divided into two classes, one for verifying requirements and one for validating product suitability. The former is based on a decomposition of product and environment, which allows targeted and structured testing even of smaller units. This selection essentially already takes place in the test planning and is transferred to the test orchestration in terms of content with the functional test specification. Validation, on the other hand, pursues the goal of demonstrating the fulfillment of the intended purpose of the overall system in all possible situations. This results in constraints that the test object represents the ADS equipped vehicle, the test is conducted on public roads and the test scenarios must be random and versatile. In this case, the test objectives are not provided by requirements, but are derived from validation criteria and are specified as part of the functional test specification.

2.2.1.2 Technical Design

Functions can only be realized with the concrete technical solution. Executable tests that are to be described in a technical test specification are therefore only possible if the technical design, i.e. the technical architecture and the technical system requirements are known and provided as input to the test orchestration.

2.2.1.3 Physical Design

The physical design provides the information when and in which configurations integrated systems are available. It thus explicitly refers to the integration of concrete HW samples and SW versions into system configurations.

2.2.1.4 Test Platforms

Another necessary input into the test orchestration is the information on available test platforms. Test platforms are understood to be SiL platforms, HiL platforms, etc., but also test vehicles on the test site or in real traffic. The decisive factor is which physical test objects can be instantiated on the test platform and which scenarios can be realized. It is also important which significance can be achieved. Above all, this requires validated test equipment and knowledge of its operational limits.

2.2.1.5 Technical Concerns and logical scenario instances

Technical concerns involve indications of a subjective (e.g. assessments) or objective (e.g. measurement results) nature, which must be given special consideration with regard to the scope of the test. In contrast to test planning, they concern the technical design and are therefore to be taken up in test orchestration. In addition, the relevant test scenarios must be derived from the total set of all logical scenario instances representing the ODD in which the technical concerns play a role at all.

2.2.2 Process and Methods of Test Orchestration

2.2.2.1 Core Process

The main purpose of the test orchestration is to generate technical test specifications that define the physical test object, the physical measured variables, and the test instance to be used. Figure 2 depicts the core process.

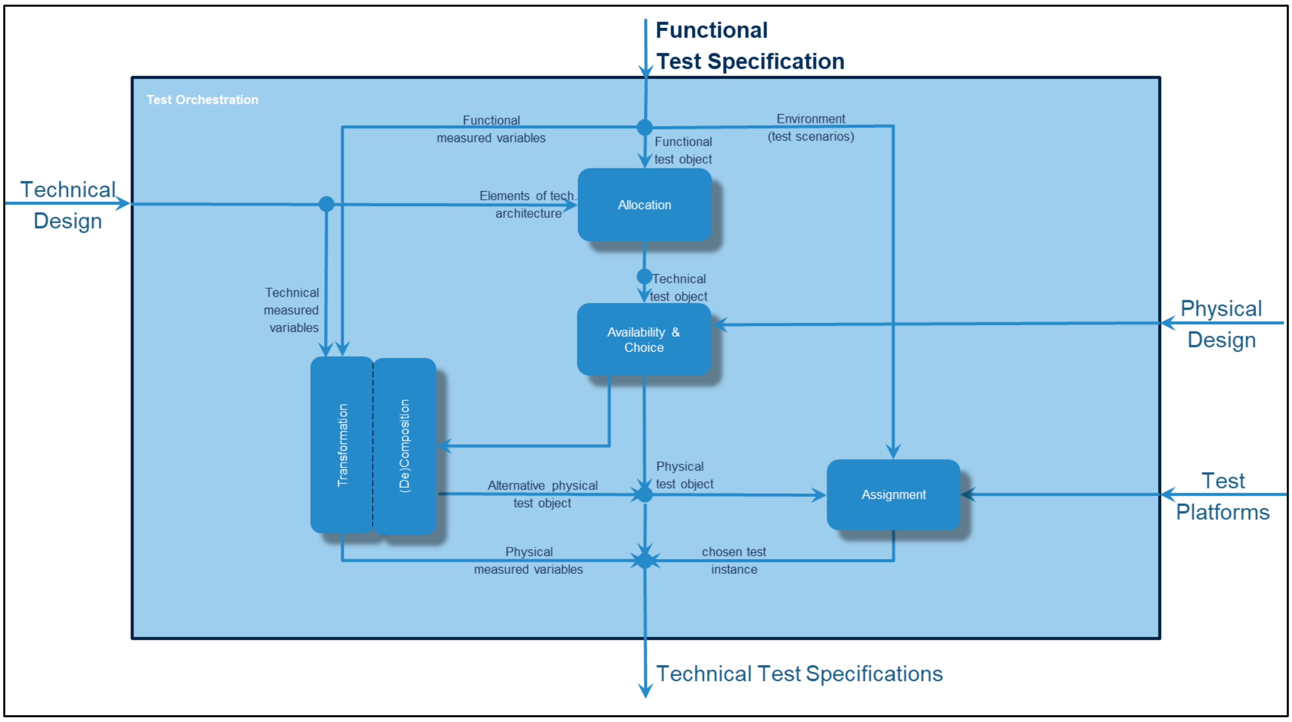


Figure 2: Test Orchestration - Core Process

Allocation

The technical test object can be identified based on the specified functional test object by tracking the allocation of the functional architectural elements to the technical architectural elements from the design process.

Transformation

The technical design with the linked requirements for the respective elements provides the specified quality and thus the total quantity of all conceivable technical key performance indicators. Using the so-called transformation, the technical parameters to be investigated are extracted from this on the basis of the functional key performance indicators, i.e. the functional measured variables are translated into the concrete physical measured variables.

Availability & Choice

The technical test object, which is available in the form of a description of a kind of theoretical model, is used to check the availability and suitability of the physical test object that is suitable and implemented according to the configuration.

Assignment

Subsequently, it is analyzed on which test platform the test should be carried out. This assignment is based on the consideration of criteria such as feasibility, effort, cost, availability and validity. Physical test object and test scenarios are instantiated on the chosen test platform, thus building the test instance.

2.2.2.2 Complete Process

The complete process is complemented by additional artefacts as depicted in Figure 3.

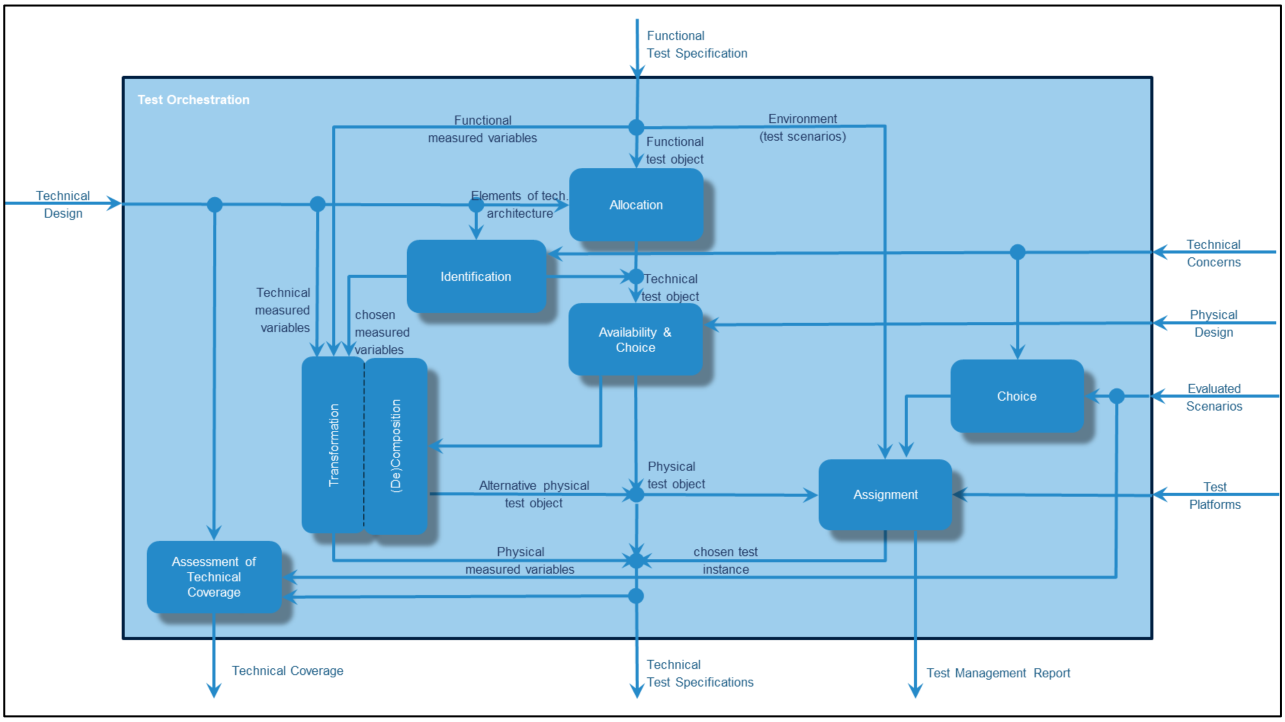


Figure 3: Test Orchestration - Complete Process

Transformation and (De)Composition

A special case exists if the technical test object is deviated from for certain reasons. This could be necessary, for example, if the interfaces for measuring the test variables are not accessible. Then the test must be shifted to another system level. Either you go one level lower (shown as "decomposition" in the figure) or you test the desired object at a higher system level (called "composition"). Both have advantages and disadvantages and must be decided individually. For example, "decomposition", i.e. decomposing the test object into several sub-objects, leads to an increase in the number of tests and potential emergences from the mutual influence of the sub-objects are not recorded if they are not taken into account elsewhere and thus secured. Tests at higher integrated levels, on the other hand, complicate the analysis of the cause-effect relationships, among other things. Regardless of the type of shifting the test, the original technical key performance indicators must be transformed to the alternative test object in both cases.

Technical Concerns – Identification and Choice of evaluated scenarios

The technical concerns require further separate treatment. This involves indications of a subjective (e.g., assessments) or objective (e.g., measurement results) nature, which must be given special consideration with regard to the scope of the test. In contrast to test planning, they concern the technical design and are therefore to be taken up in test orchestration, since only here the affected technical elements and their key performance indicators to be selected accordingly for testing these concerns can be identified on the basis of the technical design. In addition, the relevant test scenarios must be derived from the total set of all logical scenario instances representing the ODD in which the technical concerns play a role at all. Thus, the technical concerns lead to their own additional test cases, which complement the top-down structured approach in test planning to cover design and ODD. Accordingly, a separate technical test coverage is determined for the test orchestration.

2.2.3 Outputs of Test Orchestration

2.2.3.1 Test specification

In the overall process of VV methods, the technical test specification (TT) represents the exchange document between test orchestration and test execution. The TT thus acts as a communication line between test managers and test engineers. The format and the implementation of the TT have high goals. On the one hand, it should enable the consistency of the safety argumentation across the test platforms SiL, HiL, test site and real traffic. In terms of test automation, the TT must therefore be at least partially readable and interpretable by both humans and machines. This pervasiveness opens the gates for a redeployment of the necessary tests for V&V. Testing is moving away from costly and time-intensive, such as in real-world traffic, to massive use of SiL, especially holistic simulations of perceptual, motion planning, and actuation in real-world environments. On the other hand, the TT has to join a long history of different formats and concepts for test description from diverse project partners, from previous research projects like the PEGASUS project, and from existing standards and norms like the ISTQB®, OSI, FMU, OpenX.

The TT shall be applicable to diverse types of tests

- Component and system testing
- Scenario-based and non-scenario-based testing
- Open-loop and closed-loop testing

Against this background, we worked out a suggestion for the format of the TT and successfully applied it exemplarily to the four test platforms.

Format of the technical test specification

In various multilateral discussion groups, the following paradigms for the creation of the TT were developed for VV methods:

- The test engineers must already have sufficient information available purely through the TT to perform a test with a predefined test objective.
- The TT describes only one test instance, since typically an engineer has expertise in only one test instance. For direct comparison measurements of different test instances, such as a simulation in SiL and a test in real traffic, two analog TTs shall be prepared.
- The TT contains at least one test case, which can be extended to up to N test cases by specified variation rules from the functional test specification. Each individual test case then provides measurement data, which can be evaluated collectively in the test evaluation.
- The test evaluation is basically part of the TT and should reflect the test objectives.
- In principle, individual fields in specific TT can remain empty. This allows for diverse derivates of the TTs in one format.

In order to fulfill the requirement of machine readability, we propose a JSON format (JavaScript Object Notation), which is already used by individual partners in the project to describe tests. JSON is an open standard file format and data exchange format that uses human-readable text to store and transmit data objects. It is a common data format with diverse applications in electronic data interchange, including web applications with servers. Thus, it is a promising candidate for future interfacing with the diverse interfaces of current and future test instances.

Overview

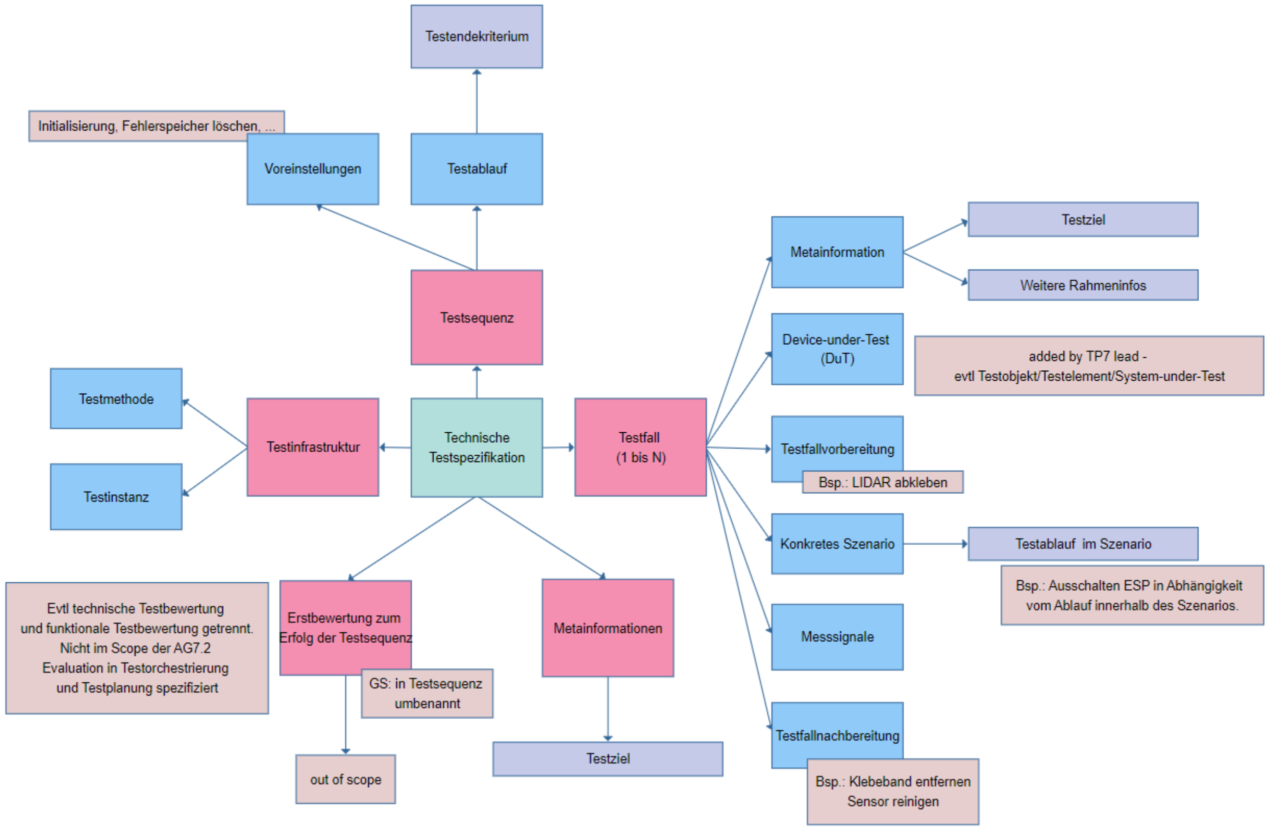


Figure 4: Hierarchical representation of the chapters of the technical test specification.

The chapter structure of the TT is shown in Figure 4 according to the status of the work. It is divided into five main chapters: Metainformation, test case, test infrastructure, test sequence and Initial Evaluation on the Success of the Test Sequence. The main chapters (red boxes) are assigned up to two further levels of subchapters. The evaluation of test data and the creation of collected results of a test is explicitly not described in the TT.

Metainformation

In this chapter meta information for the entire TT is stored. This is, for example, the test target for the entire TT or also the change history of the document.

Test case

The test case represents a single, executable test with static and dynamic scenarios. The test case contains the following subsections

1. Metainformation: Contains additional information for the test engineer such as the test goal.
2. Concrete scenario: Derived from logical scenarios as defined by the PEGASUS Project.
3. Device-under-Test (DuT)
4. Test case preparation: Initial preparations before starting a single test case.

5. Measured signals: List of signals for recording such as vehicle signals or external reference signals (e.g. time-to-collision).

6. Test case follow-up: Procedure to be conducted after the test such as bringing the test instance to the initial state.

The first subchapter "Metainformation" with the test objective for a single test case and the second subchapter "Concrete Scenario" according to the definition from the PEGASUS project form the core points of the test case.

Test infrastructure

The test infrastructure describes the test equipment with hardware and software, which does not depend on the individual test case. The test infrastructure includes the following subsections

1. Test instance configuration: Concrete description of the (minimal) hardware and (minimal) software, e.g. models, of the test instruments for test execution.

2. Test data management: Advice on how to store the test data.

The main focus lies on the test set, which describes in detail the required artifacts, in particular concrete hardware and software, with corresponding version numbers for the required traceability. The parameters for performing tests on a test instance are presented in the chapter "Test case".

Test sequence

The test sequence contains the prescription for the execution of individual test cases. The test sequence contains the following subsections

1. Test sequence preparation: Initial settings for running the test sequence.

2. Test sequence: Process description for executing the individual test cases. Serial or parallel are options here.

The key point here is the test sequence, which describes in detail each step to perform the test for the test engineers. For automated execution, this section is at least partially machine-readable.

Initial Evaluation on the Success of the Test Sequence

This chapter explains an initial plausibility check of the test performed by the test engineer. The purpose of this procedure is to detect gross errors in the test execution at an early stage. No complete evaluation of the tests is described.

2.2.3.2 Monitoring test coverage

The main purpose of test orchestration is to enable test execution by providing technical test specifications. During this process, it must be monitored to what extent the technical design can be verified and validated by the technical test specifications and which gaps still exist. The result of this monitoring serves as input for the argumentation.

2.2.3.3 Testmanagement Report

The test management report documents assumptions and decisions made in the test orchestration. It also discusses the feasibility, efficiency and timing of tests.

2.3 Test Execution

By providing technical test specifications, test orchestration enables seamless testing, meaning the realization of complementarity, efficiency, reusability, consistency and flexibility within test execution.

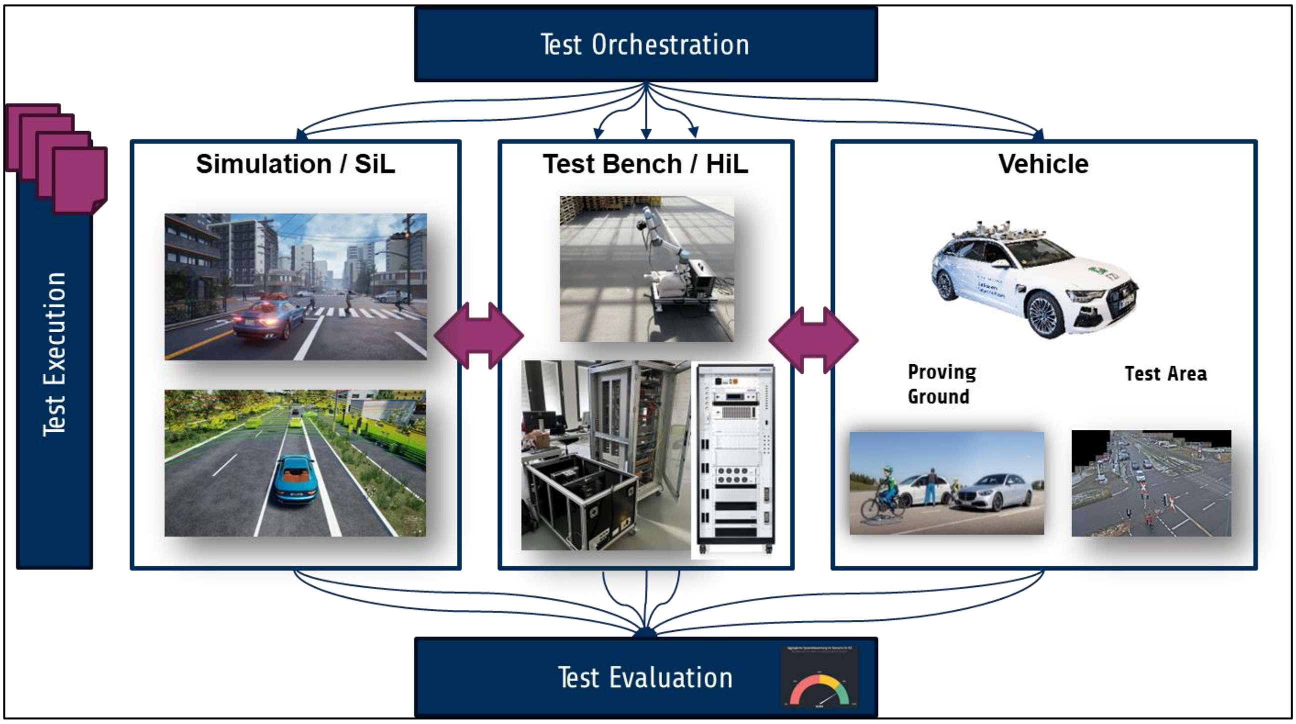


Figure 5: Realizing seamless testing

Chapter 3 on Test Execution explains seamless testing and shows example realizations contributing to the validation of the test orchestration process.

2.4 Test Evaluation

Adapted to the 2-step procedure for functional and technical planning of tests, evaluation is also divided into a functional and a technical step (see Figure 6). The technical evaluation corresponds to test orchestration. On this level, the test data generated from the execution of the tests is technically evaluated. Since the technical interfaces and signals do not necessarily correspond to the functional ones, a reverse transformation of the technical measured values into functional ones is also necessary, analogous to the step from the functional to the technical specification. Thus, the approach of architecture, design, and test planning to separate functional and technical aspects with all its advantages is also consistently pursued for the evaluation.

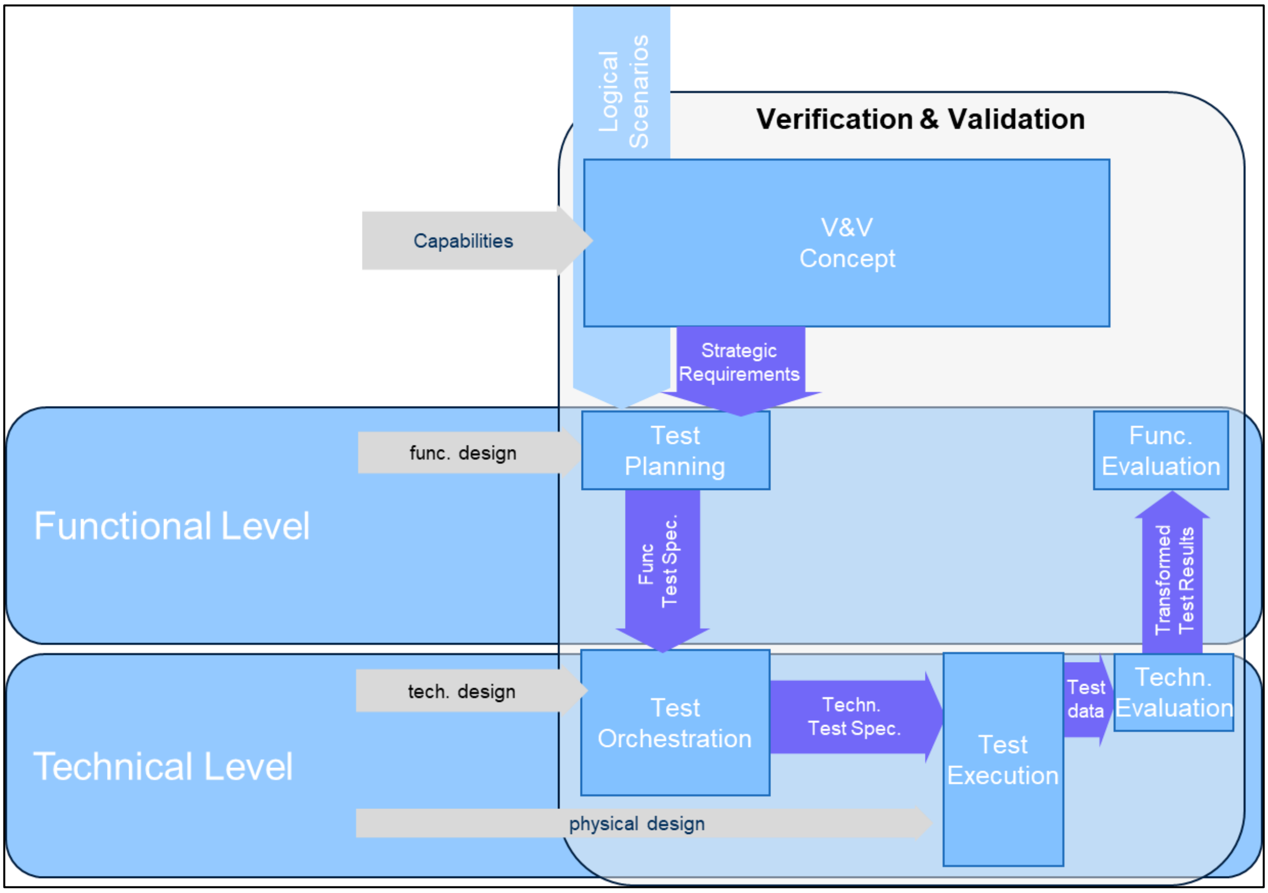


Figure 6: Technical & Functional Evaluation in VVM

Up to this point, the described evaluation still deals with individual test cases. These test cases arose from decomposition in three directions:

- Test object: Product System -> Components
- Test operational environment: ODD -> Scenarios
- Test objectives: Top Safety Goal -> Safety Sub Goals

The final goal is a holistic statement on safety, i.e. completely with regard to the product, its operational environment and the safety objectives pursued. In short, the individual partial results of the decomposed test cases must then be merged again into an overall result. Figure 7 depicts an example of the challenge of aggregating multiple functional results.

In other words, the individual contributions to uncertainty are "added up" over the sum of the components, scenarios, and safety objectives. This is done in an aggregation step within the Functional Evaluation. The consolidation of the evaluation uses the same model according to which the safety requirements were previously broken down to components, scenarios, and sub-goals, and which is in the "Risk Management" block in the Assurance Framework.

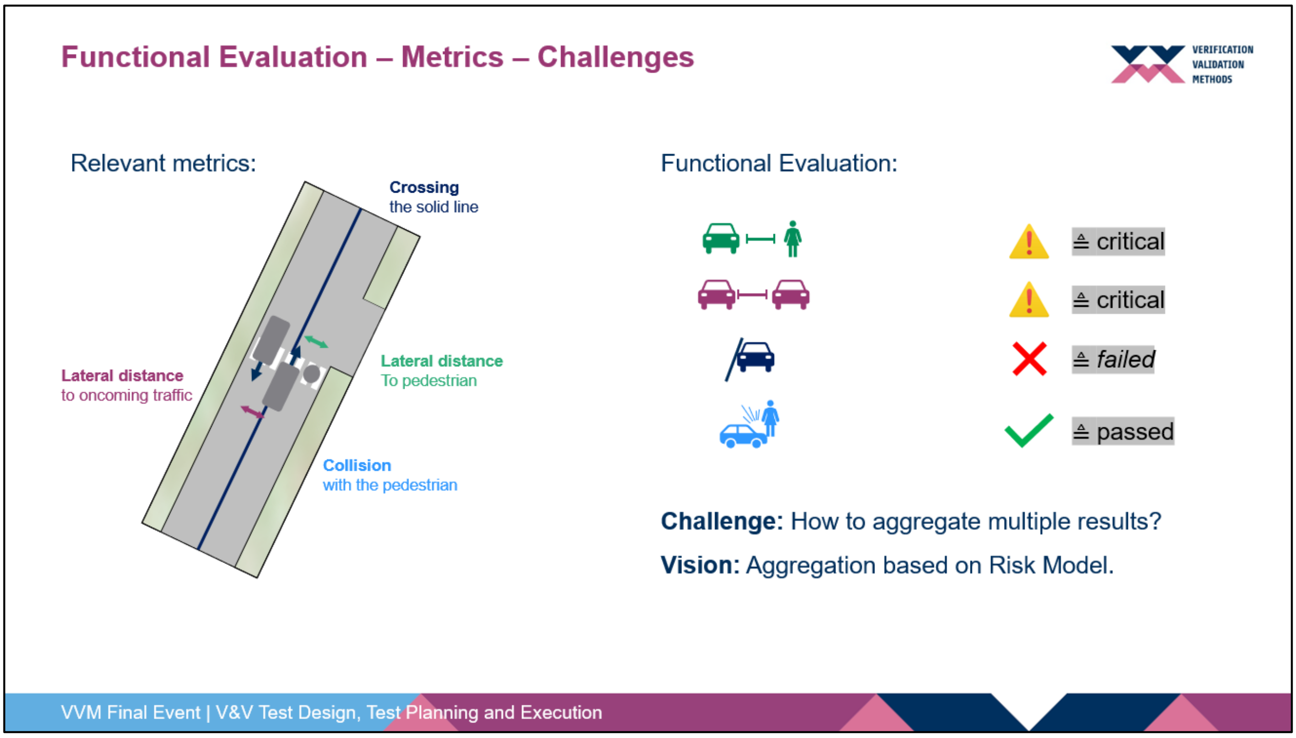


Figure 7: Example on multiple results and the challenge of aggregation

Thus, the consolidated evaluation provides a significant part of the argumentation, namely the objective evidence for the system behavior of the ADS product derived from the application of the methods in the interlinked process steps.

3 Test Implementation

3.1 Seamless Testing

The term seamless testing comprises the complementary argumentation basend on the test process presented in Section 2. This should be supported also by seamless utilization of different test platforms. The following key topics with respect to seamless testing are addressed:

- **Vertical Aspects:** Seamless testing should enable a complementary argumentation reaching from test orchestration, execution to test evaluation, as shown in Figure 6. This includes breaking down from functional design & functional testing to technical design & technical test specification. The test execution is performed in parallel on different test platforms and following different technical test specifications. Afterwards, an evaluation of each test has to be performed based on different metrics, derived from the corresponding system KPIs. Additionally, all individual test results have to be harmonized with respect to the entire safety argumentation. The seamless approach should support a harmonized and traceable
- **Horizontal Aspects:** Different test platforms are available for executing tests on a technical level. This reaches from simulation in a virtual environment through X-in-the-Loop approaches on a semi-virtual level up to vehicles tests in the real world which can be performed on proving grounds or public test areas. An overview of these test platforms is presented in **Fehler! Verweisquelle konnte nicht gefunden werden.**.

Figure 8: Different test platforms and their seamless usage

These test platforms should be utilized complementary and seamlessly. One example is enabling seamless test execution of specific tests or scenarios on different test platforms, by using the same scenario description, same models or same metrics / KPIs. Another aspect addresses the comparability of test results from different test platforms to enable an overall argumentation.

3.2 Assessment of test platforms

The different available test platforms must be utilized complementary for the argumentation. The test orchestration ensures an optimized usage of the platforms based on the relevant inputs:

- Test objects, goals and scenarios
- Target KPIs and metrics
- Available test platform

To ensure an optimized usage, deep knowledge about the available test platforms is required. This includes different characteristics and behaviors of the test platforms to derive strength and weaknesses. An excerpt of characteristics is presented in Figure 9.

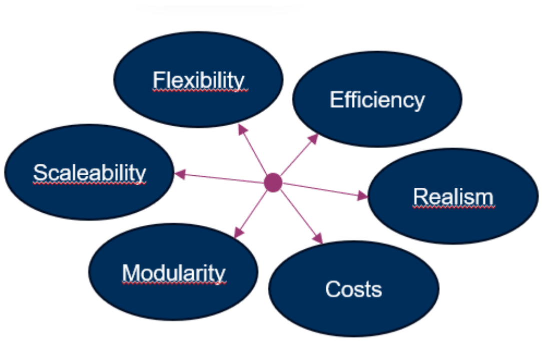


Figure 9: Different characteristics of test platforms

A systematic comparison of test platforms is performed to analyze these characteristics. As an input for the test orchestration it should enable and individual choice and combination within the test orchestration. Test platforms can be divided into real, semi-real and simulated depending on the degree of integration into the overall system. In virtual test platforms, the missing parts of the overall system are covered by simulative parts (e.g. residual bus simulation of the missing vehicle bus). The test platform describes the basic structure or approach. It should be noted that not every functionality in the vehicle can be tested on every test platform. If we address a specific platform, such as the driving assistance hardware-in-the-loop test bench, this specific version is called a test instance. Concrete scenarios can be executed on test instances.

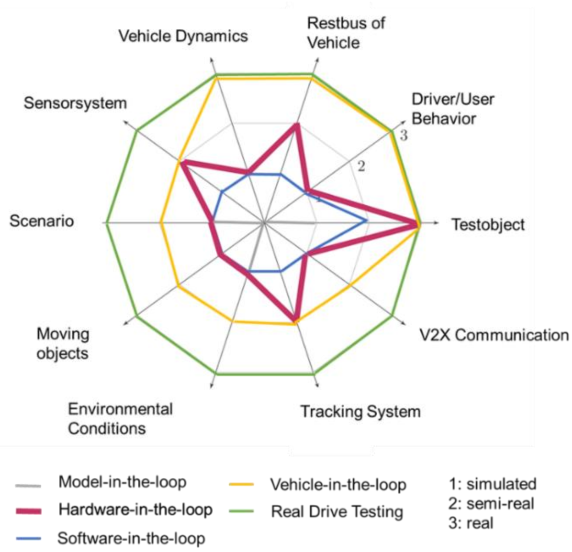


Figure 10: Comparison of different test platforms based on the degree of integration

3.2.1 Assessment criteria for test instances

Category	Hierarchy	Description	Example
Description of tests	High Level	Format of test description	OpenScenario
Description of tests	Technical Implementation	Accuracy of measureands	Accuracy of GPS-position, time resolution,...
Parameter Variation	High Level	Adjustable scenario parameters	Speed, acceleration,…
Parameter Variation	Technical Implementation	Parameter range	Speed: 10-50 km/h Acceleration: 1-5 m/s²
Parameter Variation	Technical Implementation	Parameter step sizes	Speed: 5 km/h Acceleration: 0.5 m/s²
Parameter Variation	Technical Implementation	Implemented variation algorithms	
Available ODD Elements	High Level	Available static ODD elements	Level 1-3 from the 5-Level- Model
Available ODD Elements	Technical Implementation	Class of available static ODD elements	Cars, traffic signs, buildings, traffic lights, obstacles,…
Available ODD Elements	Technical Implementation	Road Layout	Plain area, road markings, intersections, roundabouts…
Available ODD Elements	High Level	Classes of dynamic ODD elements	
Available ODD Elements	Technical Implementation	Dynamic characteristics of dynamic ODD elements	Movement patterns, inclinations, …
Available ODD Elements	Technical Implementation	Static characteristics of dynamic ODD elements	Clothing, size,…
Available ODD Elements	Technical Implementation	Property level of detail	Reflectivity of clothes, color, specific cars
Available ODD Elements	Technical Implementation	Real behavior of dynamic ODD elements	Bicyclist with leg movement, reflections from target, UFO or cable pull
Available ODD Elements	High Level	Executable maneuvers from test vehicle/steering robots	turn, lane change
Available ODD Elements	Technical Implementation	Available dynamic of the maneuvers	Speed, acceleration, lateral dynamic, steering angle
Available ODD Elements	High Level	Available weather conditions	Wind, sun, rain, clouds, fog, snow, spray,...
Available ODD Elements	Technical Implementation	Physical effects of weather conditions	

Available ODD Elements	High Level	Existing disturbance factors from the test instance	Reflections from rails of the pedestrians
Available ODD Elements	High Level	Testable phenomena	Occlusion of objects, low sun
Scope and Accuracy	Technical Implementation	Accuracy of test execution	Deviation from specified trajectory - Deviation of mean value 1/(N) * sum((x_i)) to reference value (if reference value is available) - Std. deviation of the measured values: 1/sqrt(N) * sqrt(sum((x_i - x_ref))^2)
Scope and Accuracy	Technical Implementation	Repetition accuracy of test execution	Deviation from specified trajectory for multiple runs - Std. deviation of the measured values from one run to the mean value of all 1/sqrt(N-1) * sqrt(sum((x_i - <x>))^2)
Effort for test execution	Technical Implementation	Financial effort for generation of test cases	
Effort for test execution	Technical Implementation	Time effort for the generation of test cases	
Effort for test execution	Technical Implementation	Financial effort for execution of test cases	
Effort for test execution	Technical Implementation	Time effort for execution of test cases	T_Test = T_preperation + T_Execution+T_Postprocessing
Effort for test execution	Technical Implementation	Financial effort for preparation of the test execution	Costs for the setup of the scenery (personal, buildings, other vehicles)
Effort for test execution	Technical Implementation	Time effort for preparation and post processing of the test execution	T_Test = T_preperation + T_Execution+T_Postprocessing Time effort for preparation: bring vehicles in initial position, clear error memory of ECUs, load scenarios into simulation Time effort for post processing: Measurement data storage, restore initial conditions on test instance, ...
Effort for test execution	High Level	Availability of test instance	
Demands for test execution	High Level	Required capabilities of the test object for the execution of tests	Lateral control available, certain ECU, rest vehicle simulation required
Demands for test execution	High Level	Required level of maturity for the execution of tests	For tests on public roads in real traffic

Demands for test execution	Technical Implementation	Requirements according the level of integration of the test object	Synchronized ECU communication, all components must use a specified SW version
Demands for test execution	Technical Implementation	Applicable regulatory constraints	Exemption from administrative authority with prior independent external expert report
Demands for test execution	High Level	Software or actuator-side functional restrictions or limitations	
Demands for test execution	High Level	Possible Hazards during test execution	Personal injury, damage to the vehicle, sensor technology, ...
Evaluation	High Level	Available measuring equipment	External measurement, reference sensors, in vehicle bus
Evaluation	High Level	Available measureands	Position, speed, acceleration,…
Evaluation	Technical Implementation	Kind of measureands	Vehicle signal, GPS-Position,
Evaluation	Technical Implementation	Accuracy of measureands	Accuracy of GPS position, time resolution,...
Evaluation	High Level	Format of measurements	CAN-Traces, csv, …
Evaluation	High Level	Available criticality and evaluation metrics	TTC, PET, …

3.3 Example realizations

3.3.1 Overview

Within the next chapter, the developed approaches of VVM with respect to test methodologies are presented. An overview with reference to the corresponding chapter is given in Figure 11.

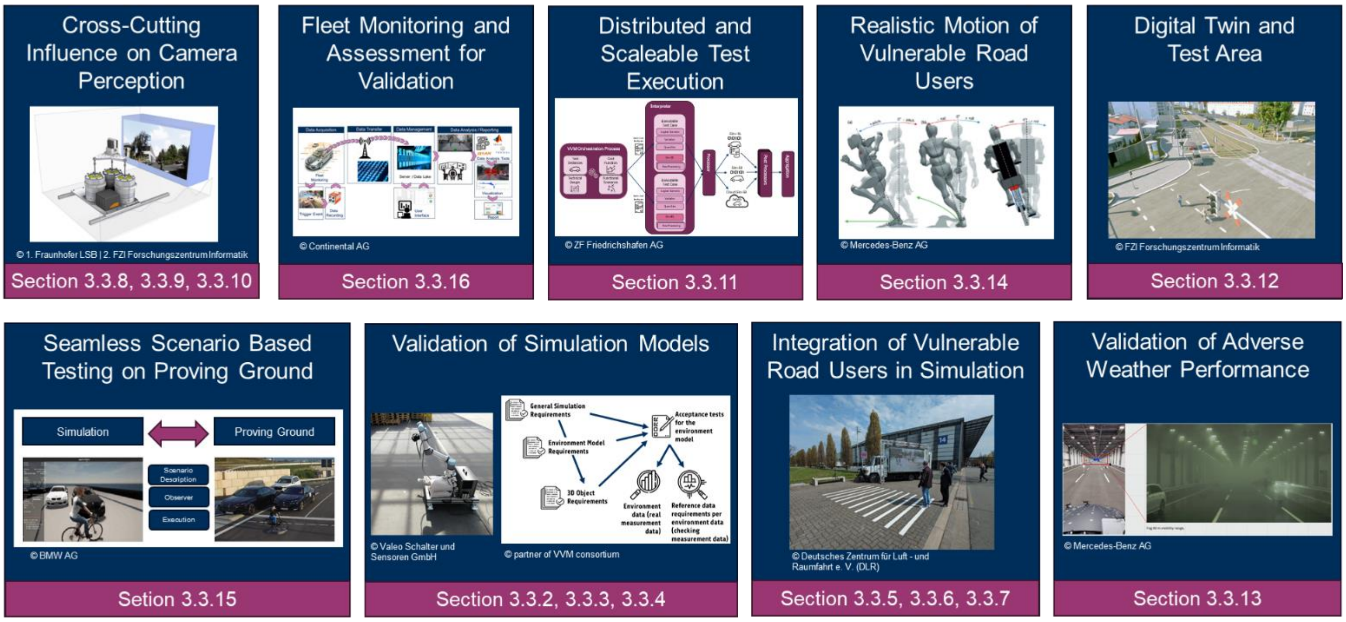


Figure 11: Overview of VVM Test Methodology Blocks

3.3.2 Cobot HiL: Measurement of LiDAR data for sensor model validation (Valeo)

Motivation

In order to use a LiDAR sensor in simulation applications, a validated model of the sensor used is required. The validation process is therefore important to generate realistic and trustworthy simulation results. This is done by comparing data from the real world with data generated from the sensor model.

Design and mode of operation

At the Cobot HiL test bench from Valeo, the LiDAR "Scala Generation 1" was used to record point clouds of asphalt at distances of 2 m and 10 m from different angles. Measurements were taken on the company's own factory premises at the Kronach site.

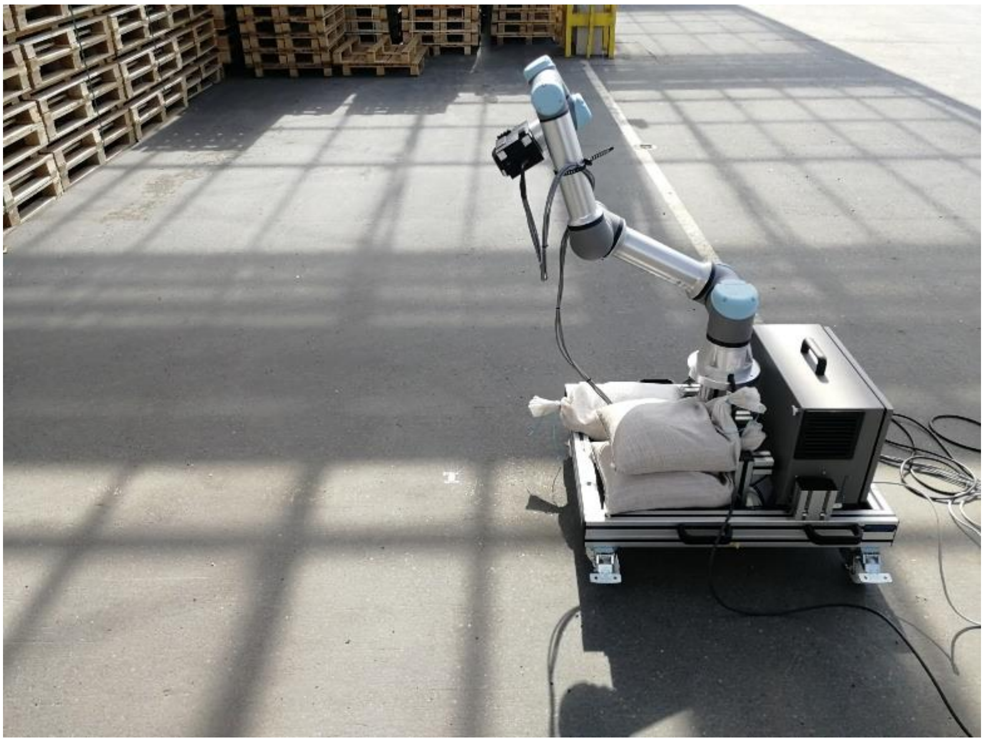


Figure 12: Cobot HiL while operating (© Valeo Schalter und Sensoren GmbH)

To realize the distances and angles, the sensor was mounted on a robot arm, thus enabling the required sensor positioning and orientation (see Figure 12). The acquisition of the LiDAR point cloud and the movement of the robot arm were controlled by a laptop. A tachymeter was used for calibration, to check the sensor position and orientation and to measure the sensor in a global coordinate system. As shown in figure 3, the LiDAR sensor has a horizontal field of view of 145° and a vertical field of view of 3.2°, divided into 4 layers.

Technical Execution

For two measurement series "2 m" and "10 m", nine sensor positions and orientations were calculated in each case so that the distance from sensor to ground was exactly 2 m and 10 m, respectively, and the sensor was aligned with the measurement point on the ground. Consequently, a circular path around the measuring point on the ground resulted for the positions. Due to the limited range of motion, the sensor positions were located in a short segment of this circular path. This is illustrated in figure 2.

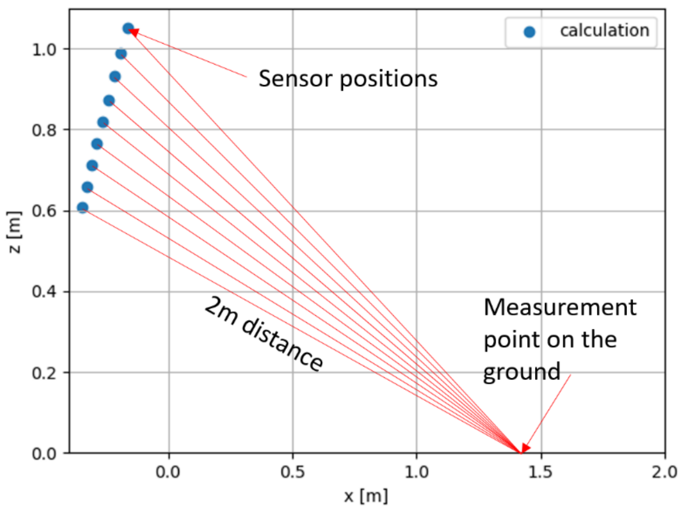


Figure 13: Sensor positions for measurements at 2 m distance (© Valeo Schalter und Sensoren GmbH)

An example of a LiDAR measurement for the distance of 2 m is demonstrated in the following Figure 14.

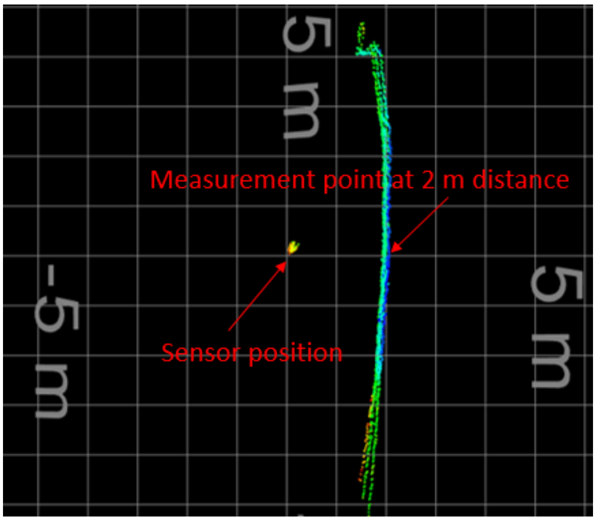


Figure 14: Visualization of point cloud data for a measurement at 2 m distance (© Valeo Schalter und Sensoren GmbH)

Integration into the VVM concept

The physical test object of the Cobot HiL is the "Scala Generation 1" LiDAR, whose physical targets are echo pulse widths and point clouds. Measurement data obtained from the HiL platform was used to validate a sensor model. The main goal of this model, at the technical level, is to reflect selected properties, especially reflectance properties of the real world. Validated models, as well as the sensor model, are used in test orchestration to be able to test different scenarios in simulations. The Cobot HiL enables an evaluation of the trustworthiness of the LiDAR model and thus contributes significantly to the validity and reliability of the sensor model.

3.3.3 Validation of an environment model (dSPACE)

Motivation

One of the main goals of VVM is to use simulation as a valid tool to test automated driving systems. To proof that the simulation is a valid tool all used simulation models should be validated. This section shows how to validate an environment model.

Requirements for a validated environment model

The following figure shows a procedure how an environment model can be validated. The procedure is grouped in 3 different abstraction levels of requirements, as shown in Figure 15.

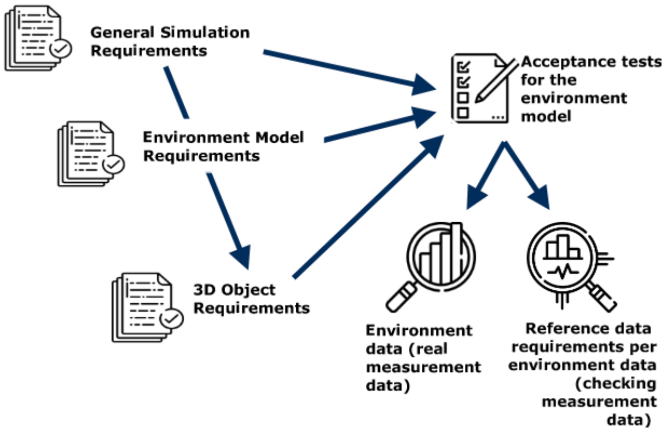


Figure 15: Validation procedure

1. General Simulation Requirements

General simulation requirements encompass various aspects such as input data, simulation models, simulation outputs, documentation, and other relevant factors that contribute to the successful execution and meaningful interpretation of simulation use cases. General simulation requirements are derived from expert knowledge, accident data, ODD, relevant standards, and regulations.

The following table shows an example for General Simulation Requirements:

ID	Description	Type
sim_req_1_1	A (global) coordinate system must be clearly defined.	General Simulation Requirements
sim_req_1_2	The simulation tool shall support OpenDRIVE 1.6.	General Simulation Requirements
sim_req_1_3	The simulation tool should generate log files.	General Simulation Requirements

2. Environment Model Requirements

The requirements for the environment model need to address the intended purpose, level of detail as well as aspects relevant for the system under validation:

- physical laws
- properties and weather conditions which may influence the sensor behavior
- traffic regulations and behavior of road users

Environment model requirements are derived dependent from general requirements, intended system architecture and system components.

The following table shows an example for Environment Model Requirements:

ID	Description	Type
sim_req_2_1	The European Road sign “crosswalk" must be present in the environmental model and capable of being represented realistically.	Environment Model Requirement

3. 3D Object Requirements

The requirements for the 3D objects ensure that the surface property, the logical properties, and the level of detail matches the requirements of the sensor:

- Wavelength range of the sensor
- Selected physical model of the surface of objects
- Level of detail of the object mesh

3D object model requirements are derived from environment requirements, intended system architecture and system components.

The following table shows an example for 3D Objects Requirements:

ID	Description	Type
sim_req_3_1	The wavelength range of the BRDF shall include the wavelength range of the sensor.	3D Object Requirements

3.3.4 Empirical Simulation Validation (DLR)

Comparing proving ground and simulation data of an SAE Level 4 System

The goal of this activity was to evaluate the degree of realism in a detailed simulation for an automated vehicle and to develop and apply a method for evaluating the validity of the simulation results.

Figure 16: Left turn scenario with driving SAE 4 automation on (top) proving ground and (bottom) within simulation (© Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR))

Approach

- Targeted experiments with an automated vehicle are performed on a proving ground.
- A model of the SAE 4 vehicle using the same automation function is run in simulation
- The recorded behavior of the other vehicles from the proving ground are replayed in the simulation
- The simulation outcome is compared with the reference data
- Discrepancies are analyzed and explained, using internal data from the test vehicle in addition to the external behavior data

Experiment Setup

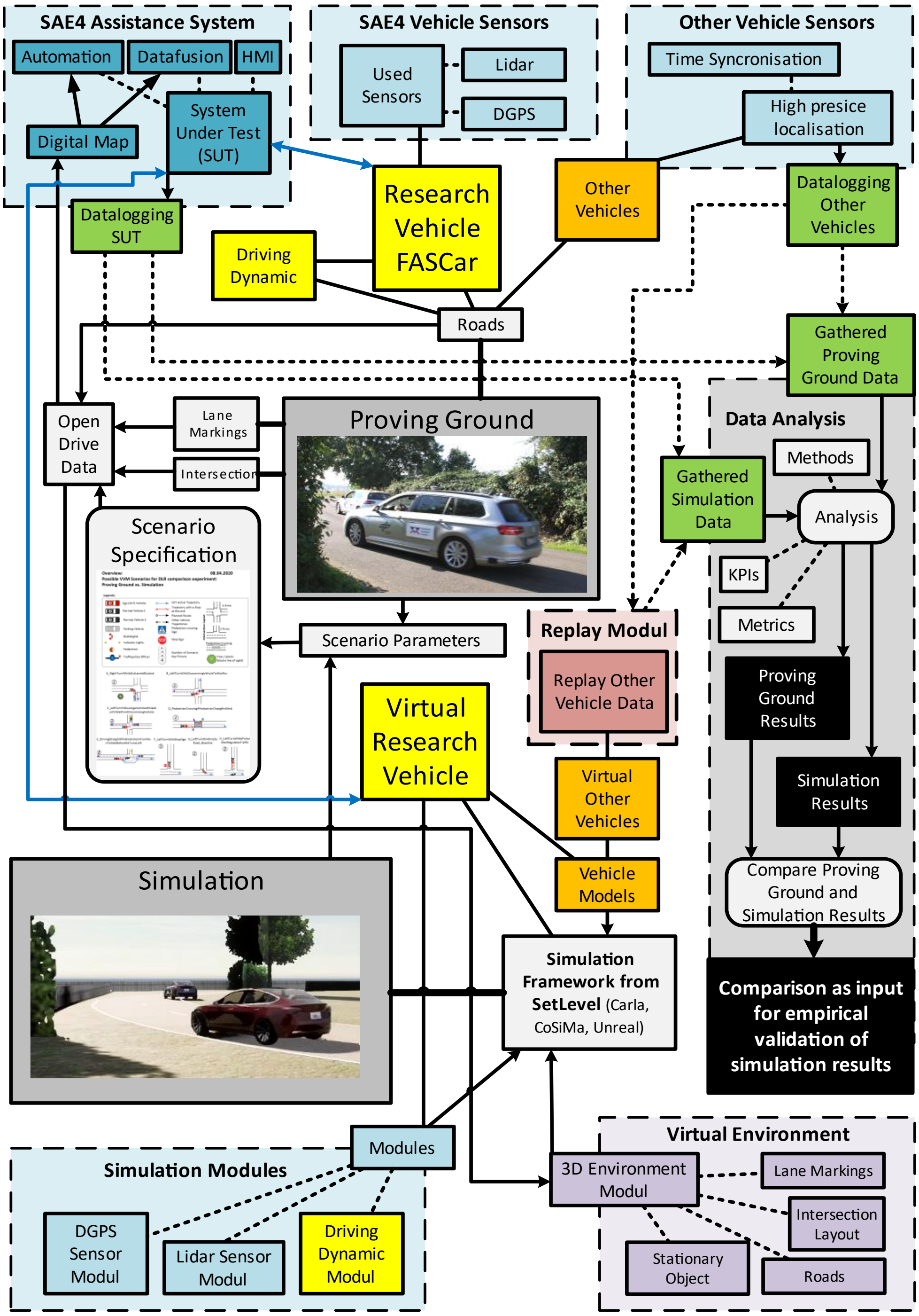


Figure 17: Components and dependencies of the experiment setup to compare and evaluate collected proving ground and simulation data (© Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR))

Addressed requirements for a comparison within experiment setup:

Both environment (proving ground and simulation) using the same scenarios and parameters, sharing the same OpenDRIVE based digital map, and having the same SAE level 4 automation as system under test in use.

Proving ground specific components are:

The research vehicle ViewCar2, which uses a LiDAR, GNSS\DGPS sensor setup. The other automated traffic realized by the research vehicle FASCarE using GNSS to follow automated a given route without object detection and reaction on the System under Test.

Figure 18: (left) ViewCar2 SAE4 SUT (right) FASCar-E as automated oncomming traffic

Simulation specific components are:

An Unreal engine, CARLA based simulation framework. The Virtual environment as a digital twin to add buildings, trees and other surrounding elements to the simulation. Modules for vehicle dynamic and GNSS, LiDAR sensors as well as a module to replay the other vehicles from the proving ground via ROS logs.

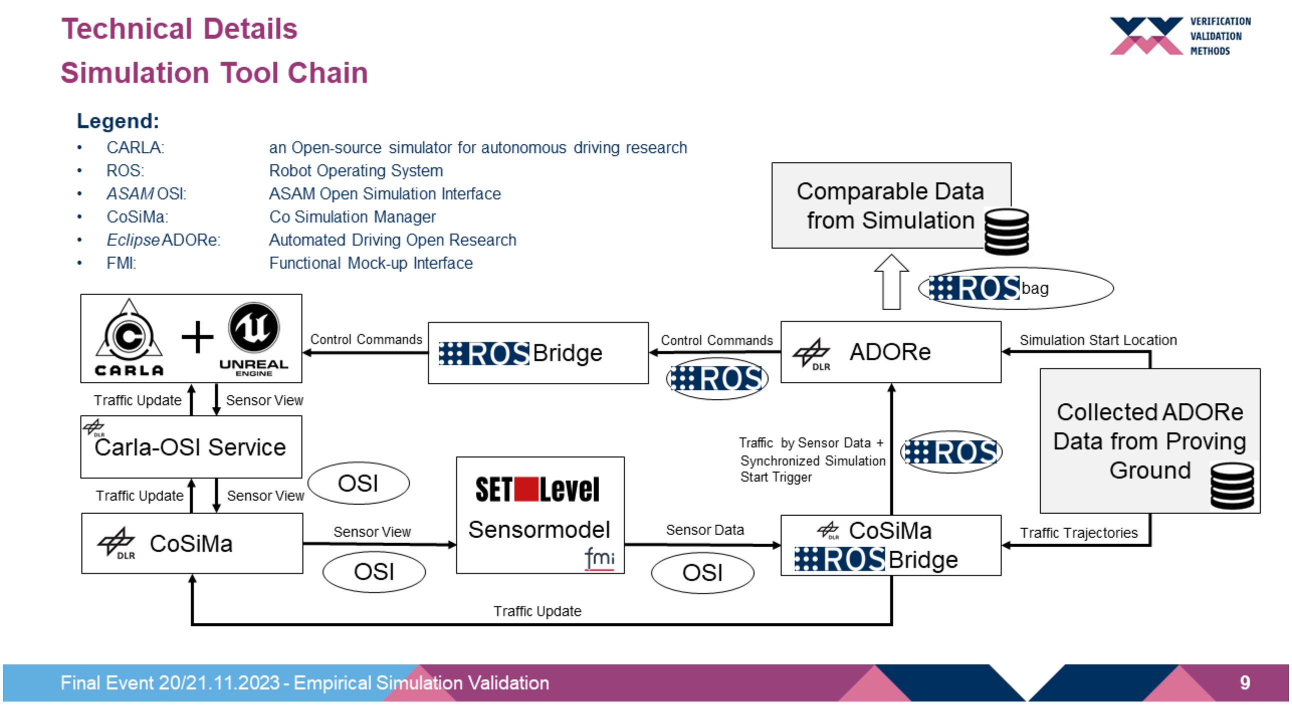


Figure 19: Overview simulation toolchain

Within the Simulation framework a combination of ROS, OSI and FMI interfaces organizes the data exchange between the replay data, the SAE4 Automation (ADORe), the sensor modules and CARLA. A Ros bag from ADORe generates simulation SUT data comparable with the logged data on proving ground.

Test Environment

As test environment a digital twin of the proving ground was used. The digital twin combines the OpenDRIVE lane descriptions with surrounding elements.

Figure 20: (left) Proving Ground in Reality viewable within Google Maps (right) Simulation bird view within Unreal engine

Addressed Scenarios

For comparable scenarios on proving ground and within simulation was a time-based start trigger used which starts the automations on both vehicles to the next full minute on the proving ground. The synchronization was logged and were reused within simulation to revise the same simulation start behavior. Logged trajectories on proving ground of the other vehicles allows a comparable behavior within simulation.

Data was collected on proving ground and regenerated within simulation for following scenarios:

- Straight driving reaching speed limit
- Follow-up journey of two automated vehicles
- Turning left without oncoming traffic at maximum speed
- Sensor test without obstacle
- Sensor test with obstacle
- Parked vehicle blocks lane
- Left turn with priority oncoming traffic

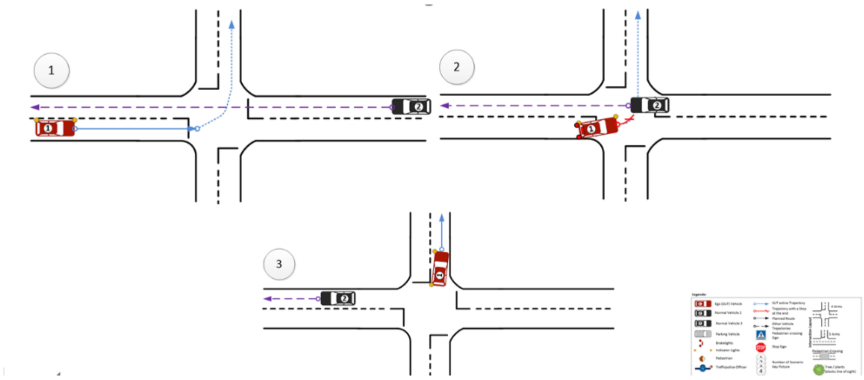


Figure 21: Showing the - left turn with priority oncoming traffic – scenario key frames

Results - Scenario Example: Left-Turn

The following example shows a left turn scenario in which the automated vehicle must observe oncoming traffic.

Figure 22: Figure 3: Scenario - left turn introduced by colleded data on the proving ground, green line shows path of the SAE 4 automated vehicle, blue line of the oncomming traffic (Data © Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR), Aerial map © City *of Braunschweig, 2020)*

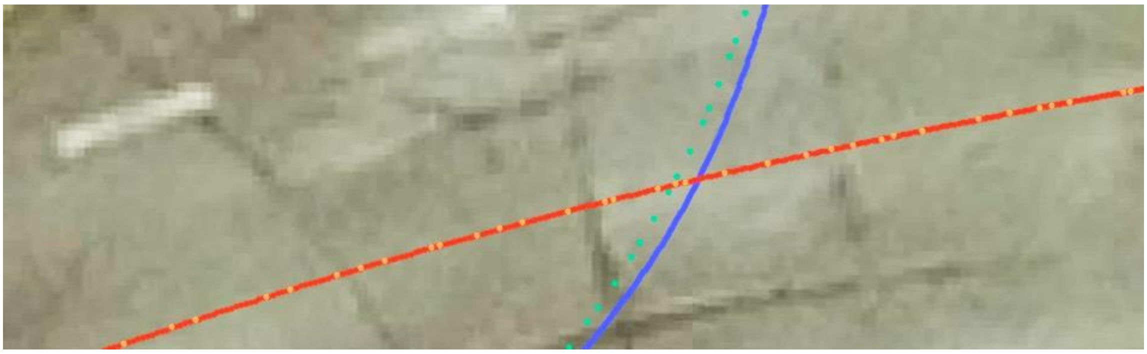


Figure 23: Detailed view on difference between proving ground (blue, red) and simulation (green, orange) trajectories within the left turn scenario (© Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR))

Observations:

- Sensor detections deviate between simulation and reality
- Trajectory of HAD-system is nearly the same
- “Decisions” of the HAD-system are the same

Velocity over time

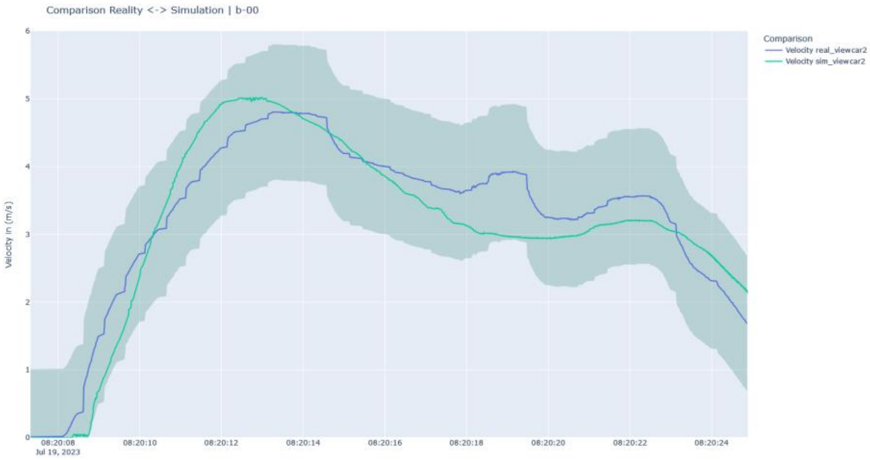


Figure 24: Velocity of HAD in simulation vs. reality

Observations:

- Overall course approximately equal
- Deviations in start, increase, variability, … between simulation and reality
- To decide whether the simulation is valid, an illustrative velocity accuracy criterion of ± 1 m/s is displayed (gray band)
- The illustrative criterion is met

Conclusions

- Simulations can be adjusted through various fine-tunings to be close to reality
- Deviations can nevertheless be observed, the measure of the maximum permissible deviation must be selected depending on the application
- Detection failures (such as ghost objects for LiDAR) are underrepresented in used sensor models

The activity demonstrates the ability of simulation to accurately replicate reality and validate SAE 4 automation systems, given that:

- Accurate (valid) models for automation components and environment are available
- Validity constraints for the models employed are observed
  - E.g. challenging environmental conditions may lower the level of significance
- The simulation has to be set up carefully
  - Validity checks under controlled conditions must be performed

Deviations will nevertheless be observed and must be taken into account in determining the level of significance of the simulation results. In particular the modeling of perception chain needs attention e.g. ghost objects for LiDAR are underrepresented in employed sensor models. The use of simulation in safety argumentations must be well integrated with other sources of factual evidence

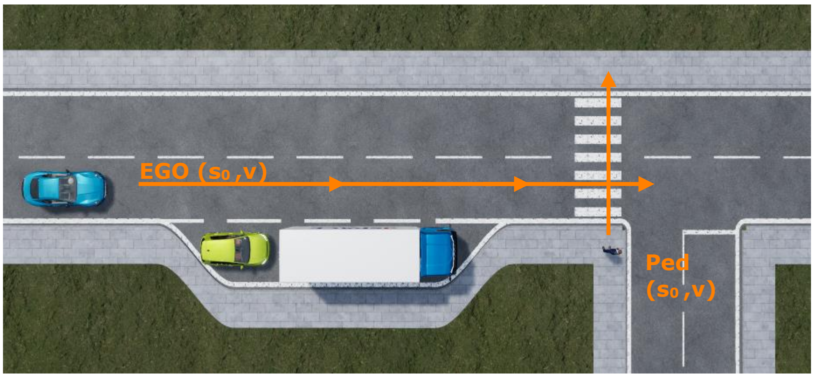
.

3.3.5 Evaluation of an AD System Using Scenario-Based Testing (dSPACE)

Motivation

One of the main goals of VVM is to use simulation as a valid tool to evaluate an automated driving (AD) system. To do this, it is necessary to test whether the AD system can handle critical scenarios. But finding a critical scenario can be a difficult task since multiple concrete scenarios can be derived from one logical scenario. One possibility is to simulate and test each concrete scenario in parallel since each concrete scenario is independent from the others. Using a computer cluster can speed up this task (scenario-based testing) and avoids the effort of finding suitable parameters that reduce the number of concrete scenarios.

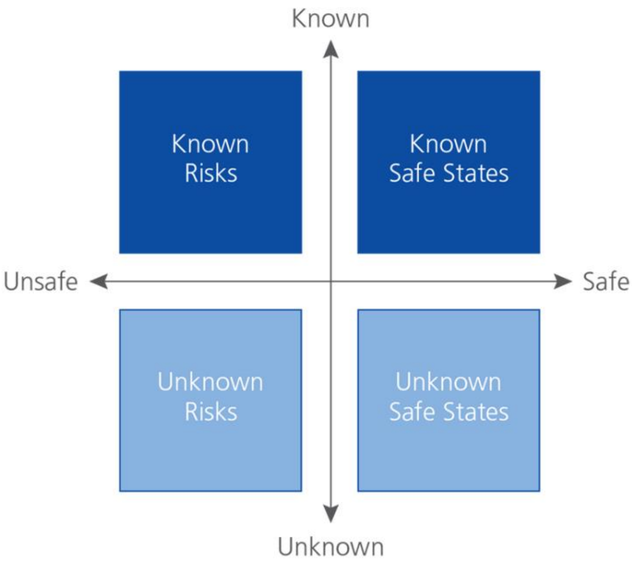
Using a Cluster-Based Solution



The previous figure shows the following logical scenario: A vehicle with an automated driving system is approaching a crosswalk. A pedestrian who is hidden by a parked car would like to cross the road using the crosswalk. To evaluate the AD system, a variety of parameters must be checked to confirm the pedestrian will not be hit.

	Ego S_0 [m]	Ego v [kph]	Ped S_0 [m]	Ped v [kph]
Min	0	0	-20	0
Max	75	60	0	10
Step size	5	5	1	1
#	15	12	20	10

The previous table shows the parameters that are considered as an example. Based on the parameters, the parameter space contains 32,000 concrete scenarios the AD system must deal with. Among this parameter space, there are known and unknown risks. Testing the whole parameter space will identify known and unknown safe states (see following figure) as well.



If the automated driving system will not hit the pedestrian in any scenario, it is safe to consider the AD system as evaluated. Using an ISO 26262-certified tool such as SIMPHERA can support that evaluation task.

3.3.6 Test execution with OSTAR (DLR)

https://www.vvm-projekt.de/fileadmin/user_upload/Final_Event/Poster/VVM_FE_Poster_14.06_Architecture_for_Configurable_Simulation_web.pdf

3.3.7 Pedestrian augmentation within simulation (DLR)

https://www.vvm-projekt.de/fileadmin/user_upload/Final_Event/Poster/VVM_FE_Poster_14.07_web.pdf

3.3.8 Hardware-in-the-Loop Test Bench (FZI)

The current assessment of environmental influences on specific components within the highly automated chain of effects is limited to advanced stages of development. Verification primarily occurs through real driving tests post full hardware integration, making it challenging to implement design optimizations during ongoing development. This approach also heightens the risk of accidents during real tests due to unforeseen performance issues. Our work aims to analyze the realization of an automated detection of system behavior of a vision component and environmental phenomena in an early stage of the development process.

Software-based simulations alone fall short in testing and capturing the full spectrum of hardware and component-interconnected limitations. Real-world test runs encompassing all relevant conditions (traffic structure, obstacles, weather, etc.) are time-consuming and yield variations that are challenging to reproduce consistently. Consequently, integrating hardware-in-the-loop on a test bench becomes essential for the future evaluation of driver assistance systems (FAS). The convergence of simulation and real-world testing on test benches is imperative to gain a comprehensive understanding of their individual strengths and weaknesses

A simulative model cannot represent all hardware constraints. A hardware-in-the-loop test bench is particularly suitable for:

- Limitation of computing power
- Shutter effects of the camera
- Time synchronicity
- Influence of disturbance variables, such as temperature and vibrations on the hardware

As part of the VV Methods project, FZI has developed a hardware-in-the-loop (HIL) test bench, the architecture is presented in Figure 25. This innovative test bench, central to scientific research, focuses on automating test processes to ensure precision and repeatability. It facilitates the coordinated execution of diverse test scenarios, guaranteeing a comprehensive evaluation of the system under test. The HIL test bench allows the creation and customization of test scenarios to simulate various operating conditions and environmental factors.

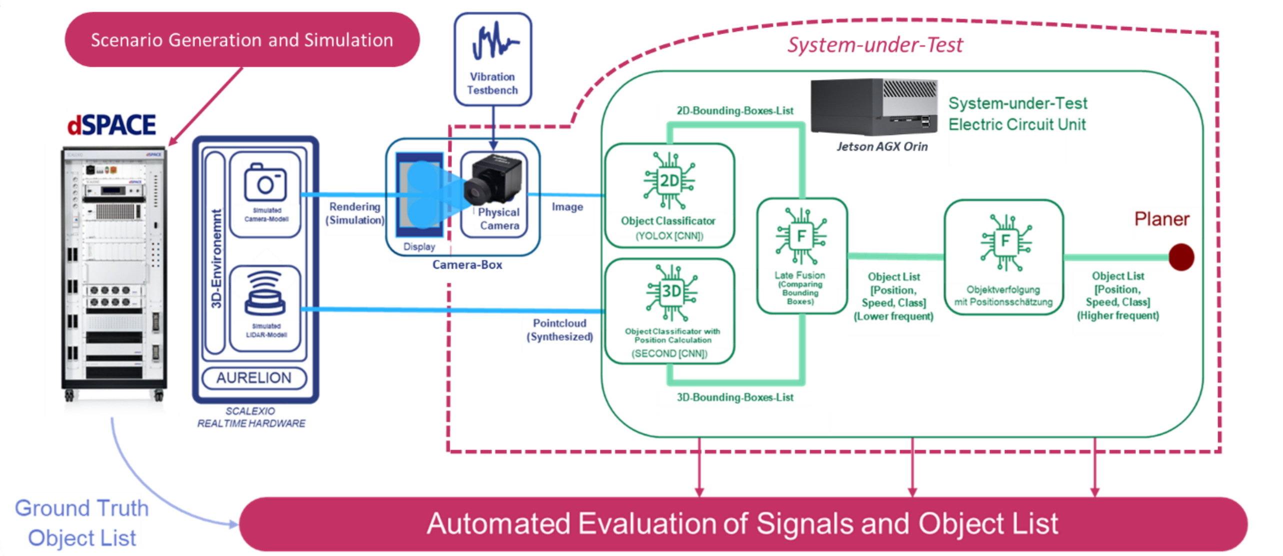


Figure 25: The cause chain of the test environment and the developed vision

Specifically designed for the automotive camera Leopard Imaging IMX-490 and its corresponding driver hardware, along with the Nvidia Jetson Embedded Hardware featuring our late Fusion Architecture of a 3D-LiDAR Detection and 2D-Image Detection, each embedded in a ROS (Robot Operating System)-Environment, the HIL test bench ensures efficient testing. It automates the initiation and monitoring of tests in both Software-in-the-Loop (SiL) and Hardware-in-the-Loop (HiL) environments, enhancing test execution efficiency. Additionally, the HIL test bench acquires and analyzes measurement data to assess the performance and reliability of the entire system under test.

Using ROS which allows to monitor all signals of the cause chain and in the functional nodes the test automation is able to detect inaccuracies and anomalies of the functions inside the test object.

During the continuous monitoring of the degradations in performance during execution of concrete scenarios, applied environmental influences and conditions are superimposable to detect cross-cutting effects and possible failure causes.

3.3.9 Schwingungsprüfstand (FhG)

Bridging the gap between simulation and in-vehicle testing

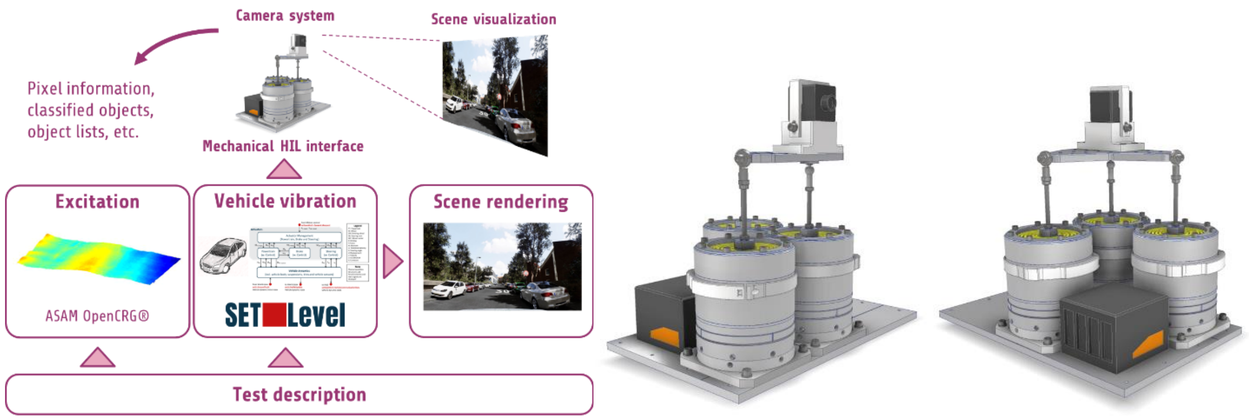
Image-based object classification depicts an integral part of highly automated driving capabilities. In this context, the verification and validation of camera-based systems might lead to new concerns. Rolling shutter, motion blur and deviations in object classification are some of the known effects caused by mechanical high-frequency excitation of camera equipment. While the emergence of vibrations at the mounting position of the camera is well understood and simulation models are available, the same isn't true for image acquisition and processing under vibrational loads. Nowadays, testing of sensitive camera equipment is performed in expensive and hard to reproduce in-vehicle tests.

VVM bridges the gap and combines highly repeatable vibration simulation with testing of camera equipment in a camera-in-the-loop test environment.

Real-time simulation of vehicle vibration

A variety of numerical models are used for the calculation of the multiaxial vibration at the mounting point of the camera. Starting with the road surface description, the numerical model integrates the ASAM OpenCRG® file format. The interaction between tire and road as well as the type of suspension, the chassis system and rigid body dynamics of the vehicle are accounted for in a non-linear reduced order MBS model whereas the latter was developed in VVM's affiliated project SET Level.

In order to account for the flexible structure of the vehicle body (i.e. bending and torsional modes) as well as vibration insulation measures of the camera (e.g. elastomer mount) the numerical model also includes a modally reduces finite element model of a body in white. Originating from the SET Level simulation philosophy, the virtual models may also be integrated in a scenario-based simulation environment in future.



Physical emulation of multiaxial vibration

A parallel kinematic is used for the emulation of multiaxial vibration. Integrating three vertical voice coil actuators with 30 mm maximum stroke and a peak force of 315 N each. The multiaxial vibration exciter approx. provides a max. vertical acceleration of 25 g, and a max. angular pitch and roll acceleration of 2800 rad/s². By making use of multiple-input-multiple-output adaptive control schemes broadband and transient vibration scenarios can be emulated with high bandwidth and precision.

Application in hardware-in-the-loop testing

Combining real-time vehicle simulation and the mechanical HIL interface, vibrations at the mounting position of sensitive sensor equipment (LIDAR, camera, gyroscope, accelerometers etc.) can be exactly reproduced while allowing for the flexible variation of mounting position, suspension settings, and road surfaces during scenario-based testing.

3.3.10 Analyzing Environmental Influences under Laboratory Conditions (FZI & FhG)

During the VVM Project the HiL-Testbench for Analyzing Perception from Section 3.3.8 and the vibration test bench from Fraunhofer LSB, see Section 3.3.9, were combined to a demonstrator to analyze environmental influences in a HiL environment. Additionally, an integration of the FZI Environmental-Simulation with procedural generation and real scanned objects from the city of Karlsruhe by rendering Scenarios on the display of the camera box is implemented.

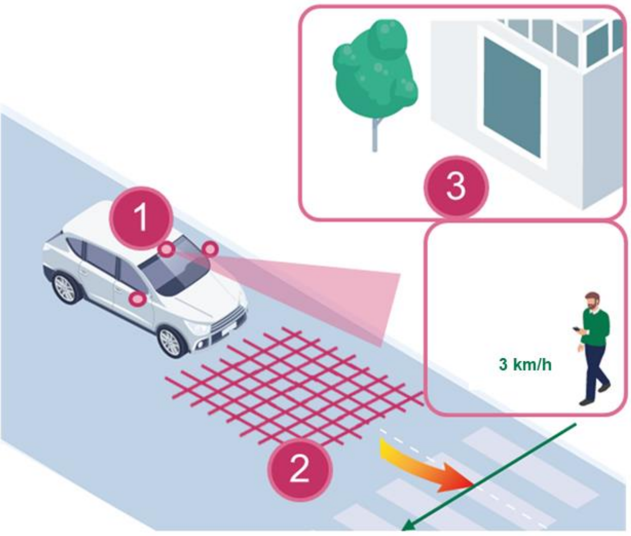


Figure 26 Realdrive conditions which are targeted to be moved under laboratory conditions

The approach is to show how real environmental effects can be transferred for the camera object detection testing. In the current state, environmental influences on individual components of the highly automated chain of effects can only be tested at a very high level of maturity. With full hardware integration, verifications are performed in real driving tests to determine any environment-related limitations. This means that optimizing design decisions cannot be implemented during the ongoing development process. In addition, the risk is increased that accidents occur during a real test due to an unexpected performance deficit. This demonstrator shows an early detection of dependencies between decomposed states and phenomena of the environment in relation to the performance of the hardware, with this focus:

- Testing the camera hardware by means of a camera box
- The laboratory application of accelerations on a camera with a moving coil system
- The procedural variation of the environment on all road layers

As a task to realize an exemplary implementation of a test instance in scenario-based testing, the vibration test bench is specified as an cause effect chain hardware-in-the-loop test platform, as shown in Figure 27. The test bench provides information about the kinematic coupling (cross cutting) to a concrete perception mode (camera) by a systematic analysis.

Figure 27 *: Applying recorded acceleration on a test camera filming procedural simulation renderings to evaluate the accuracy of an object detection*

The input is the acceleration of the vehicle motion at speeds between 10 to 30 km/h on a road profile (cobblestones with 25 cm wide stones).

The test rig allows to vary the entire kinematic frequency space coupled from the vehicle to the camera. It is possible to determine through the frequency spectrum initially independent of the rest of the environments local performance minima by capturing a continuous natural frequency. The test space can be greatly reduced by this through an early systematic detection of a generic statement: performance degradation as a function of the frequency spectrum.

Our observing target value is the accuracy of the image-based object detector under test. In addition, the environment can be varied using procedural generation to exclude the environment as an influencing factor and thus enable cross-validation for the frequency statement or finding triggering objects.

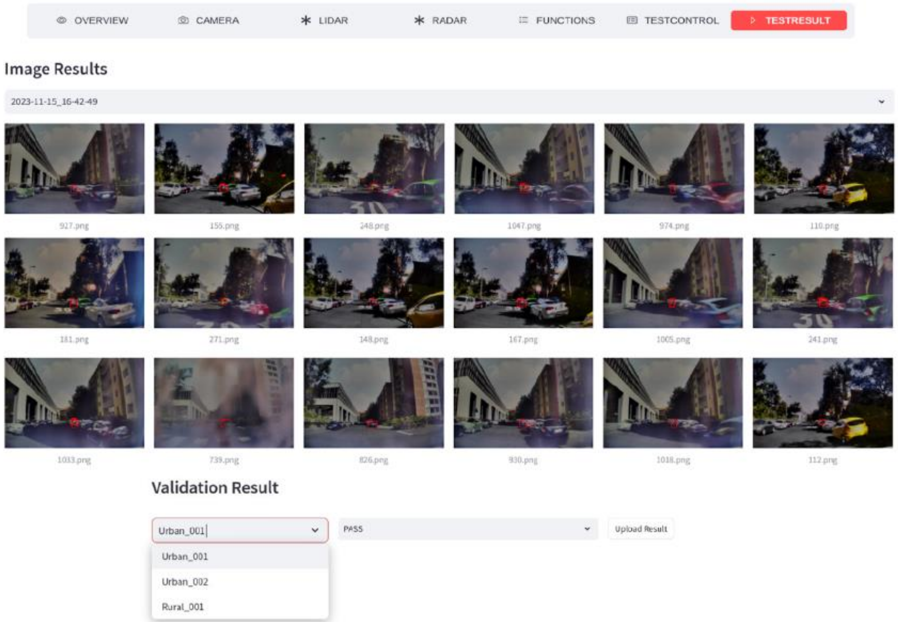


Figure 28 *: Dashboard embedded in the ROS-based test environment to configure the test object, execute the automated test instance and evaluation*

3.3.11 Distributed Test Execution and Cloud-Sim-SiL (ZF)

Orchestrated seamless testing

The Test Orchestration is defined as a process that identifies a test setup that should be used for concretizing a functional test specification by evaluating the technical design of an ADS, availability of suitable assets per test instance and cost-functions. The result of this process is the technical test specification. In order to automate the V&V-testing-process, a machine-readable representation of a technical test specification is needed. Furthermore the execution of technical test specifications on different test instances must happen in a seamless and managed way.

Technical test case specification

We have developed a DSL for technical test specifications that is implemented as a JSON scheme-based domain-model (see Figure 5) that describes test cases. Each test case consists of references to a test instance, a logical scenario, a variation-rule that expresses the derivation of concrete scenarios from logical scenarios, a device-under-test, a set of replacement models, measured variables to be recorded during the test, and a description and parameterization of the simulation-engine if any.

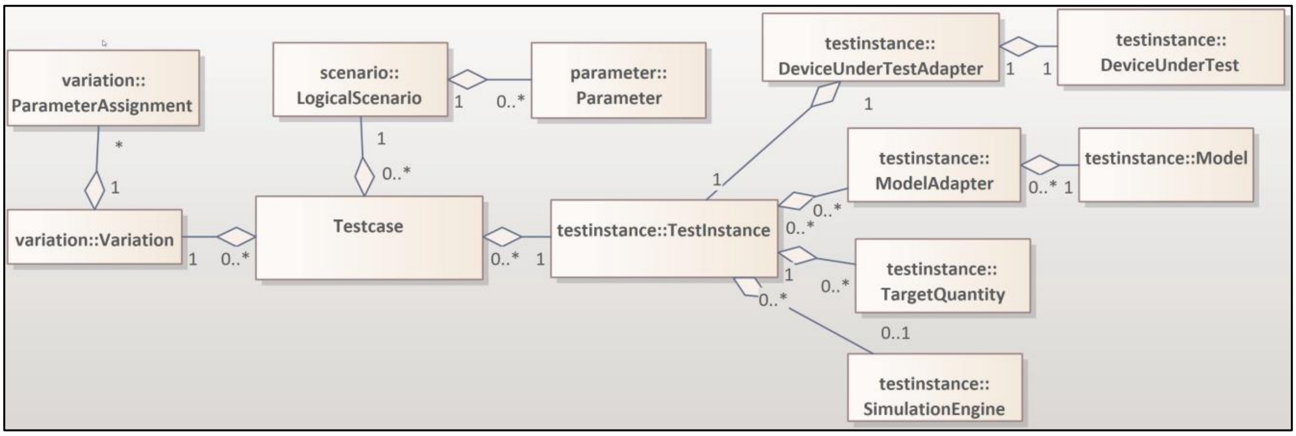


Figure 29: Domain-model of the machine-readable technical test specification

Execution framework

A test-execution-framework that interprets the incoming test specifications was implemented as a choregraphed micro-batch-processing pipeline consisting of independently scaled (post-) processors and a final aggregation step.

Generic interface for different test instances and integration into business processes

By encoding test-activities as generic technical test specifications and providing an agnostic test execution framework, we enable the orchestrated usage of different sorts of test instances that only need to be adapted to the generic test execution framework (seamless testing). Due to the event-driven architecture of the execution framework an integration into different business-processes is possible, be it the V&V process or e.g., a classical V-Model-approach.

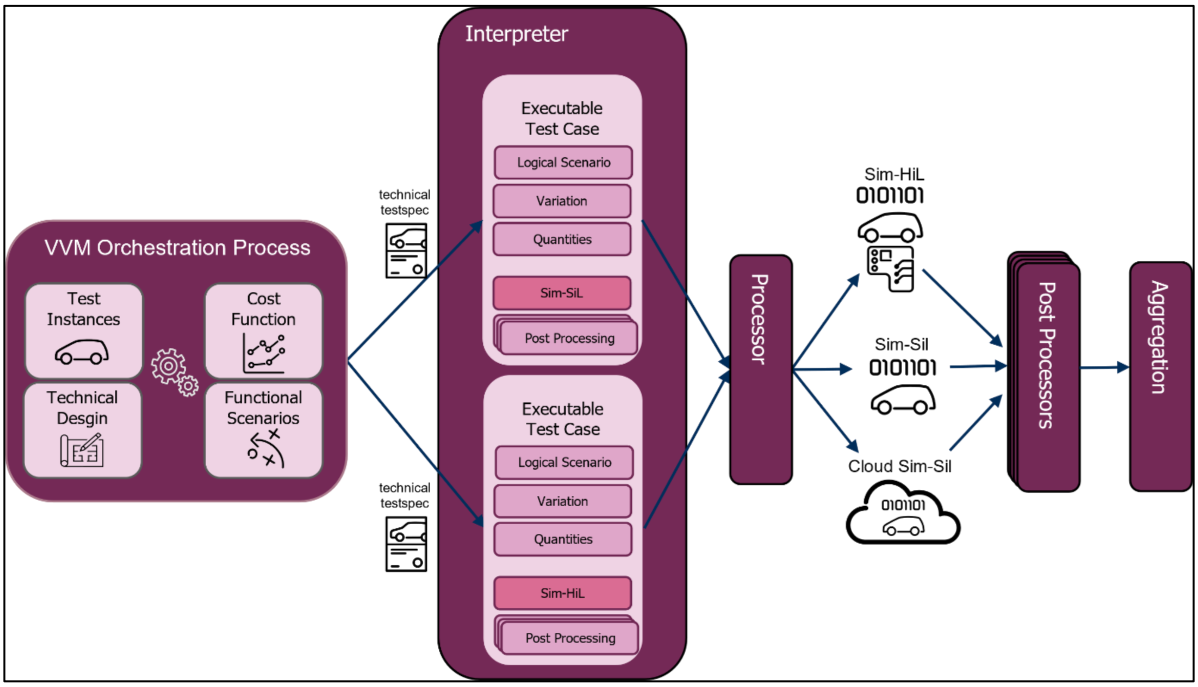


Figure 30: Execution framework

Integration of OSI-based Model

In order to verify the feasibility of integration of OSI-based models, the Reflection Based Lidar Object Model from the SetLevel-Project was used (see Figure 31).

The model is implemented using the extended SetLevel-OSI-specification. Also, the OSI-adaptation provided by IPG was used. In this setup, the model is integrated as OSMP-packaged FMU, providing detections as protobuf-parsed data-structures to the extension-hooks of the Carmaker-simulator, that are called for each cycle.

The received object-detection-data-structures could be used to stimulate the device under test.

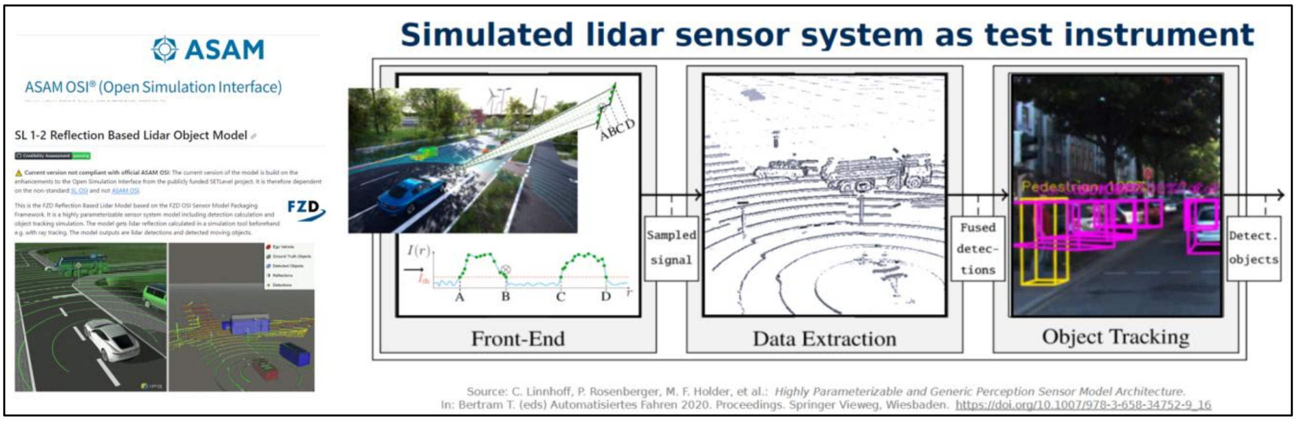


Figure 31: SetLevel Reflection Based Lidar Object Model

Integration of ZF-Driving-Function

As a Device under test, a ZF-internal driving function was integrated, by adapting IPG Carmaker to stimulate the driving-function using SomeIP-based communication as used in the Autosar adaptive based realworld target.

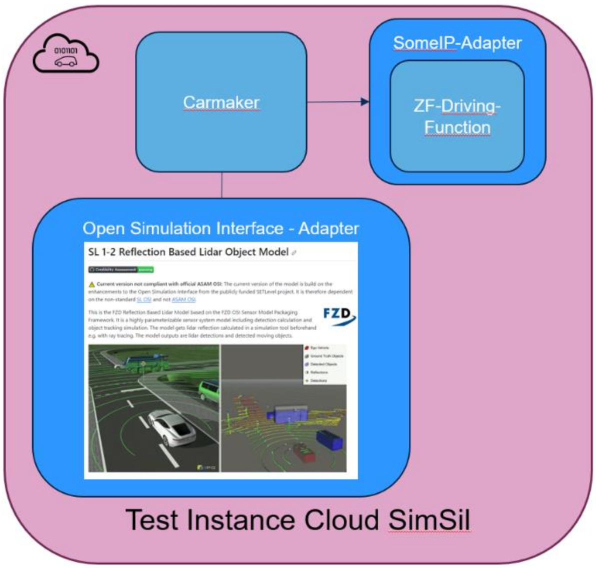


Figure 32: Simulation-Setup

Findings

The idea behind the setup (see Figure 32) was to try out the integration of OSI-based models and a real-world driving function with the proposed Execution-framework.

The integration was successful, even the correct physical transformation of OSI-received data was not completely implemented. Still, it could be shown, that OSI is a suitable model-data-exchange-mechanism, and it can be integrated with other simulation-parts that are based on different data-exchange-mechanisms like Autosar SOME/IP.

Furthermore, the high value added by using a soundly specified, machine readable Test-Specification-Format was shown, as it helps with Process-integration and auditability. Nevertheless, the sheer amount of different technical contexts, communication-techniques and model-implementations present in today’s Simulation-ecosystems provided by different vendors are still a challenge and need to be condensed further into an vendor-accepted and manageable set of specifications.

Incorporating massively scaled cloud-simulation-infrastructures are necessity for applying statistical reliability-analysis and continuous-integration as the sheer amount of needed simulation-runs cannot be executed anymore on a developers desk. The proposed event-based microbatching architecture is a good example of such cloud-based-simulation-architectures.

Nevertheless, it has to be stated, that the complexity-explosion of concrete scenarios resulting from scenario-variation can be reduced by cloud infrastructures but still exists on with smaller orders of magnitude and still has to be addressed by applying process-means as proposed by the V&V-methodology.

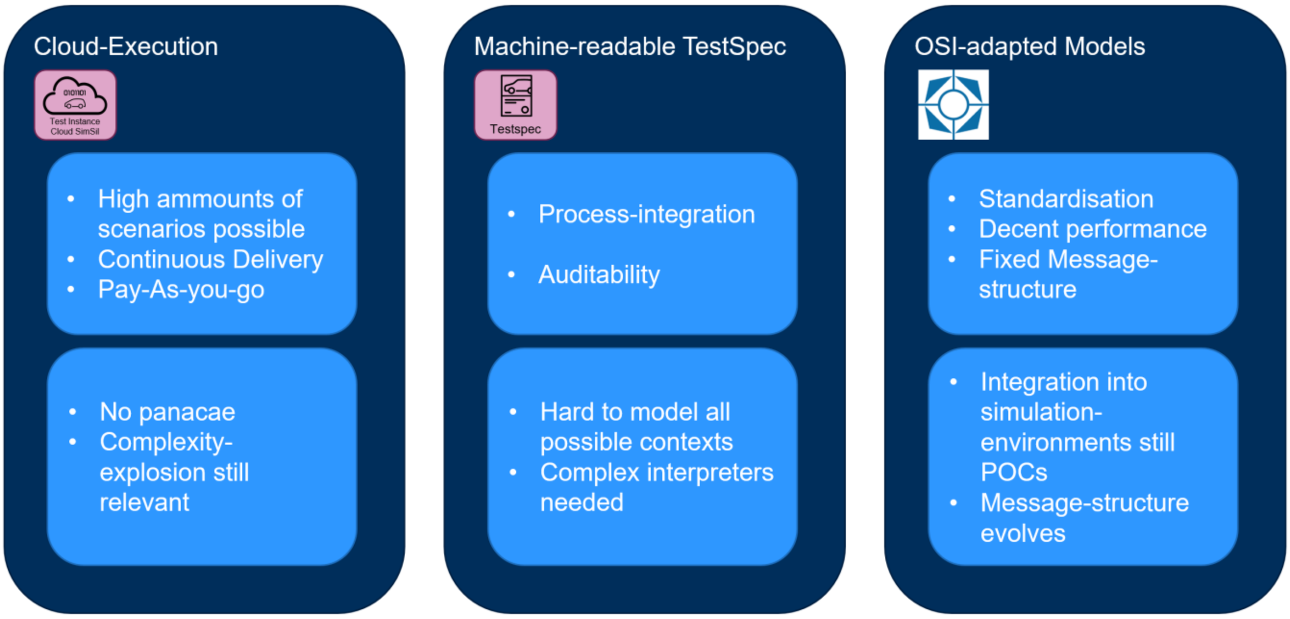


Figure 33: Findings while implementing Execution-framework with OSI-adapted model

3.3.12 Virtual Clone – Real and Virtual Test Area (FZI)

Concept

An extensive testing process is necessary for highly automated vehicles, ranging from functional testing to systematic testing under real-life driving conditions. To ensure consistency amongst these tests, which vary in their level of abstraction, they must share common interfaces and be broadly comparable, i.e. a test pipeline with different instances should fit seamlessly into each other.

Figure 34 illustrates a concept that records, analyzes and evaluates traffic data using smart test area infrastructure. With this information, specific modules of a highly automated vehicle can be evaluated in a simple open-loop simulation, which then becomes progressively more detailed and realistic in the form of a closed-loop simulation. Ultimately, the system can be tested in real-world road scenarios while simultaneously monitored using test area infrastructure.

Recording and Replay

The observation of authentic traffic scenes, including the conduct of road users and the environment in which they are situated, forms the crucial foundation for the following stages of analysis. On this basis, relevant scenarios can be identified and investigated in more detail in the simulation or real-world tests.

Simulation

High-fidelity, closed loop simulations complement real-world experiments and non-reactive log replay. The remarkable degrees of perception-, behavior- and content realism enabled through modern model based and data-driven simulations enable to scale testing to particularly rare or critical scenarios, especially including such that involve VRUs.

Test Area in Public Traffic

The test area with its infrastructure serves as a reference measurement system. This precise, calibrated measurement sensor technology provides a more profound understanding of the traffic scene that would not be available directly from the SuT.

In addition to monitoring the system under test's sensors (e.g. perception evaluation), the test area also assesses the system's overall behavior.

Evaluation metrics provide insight into how the SuT performed in the present environment.

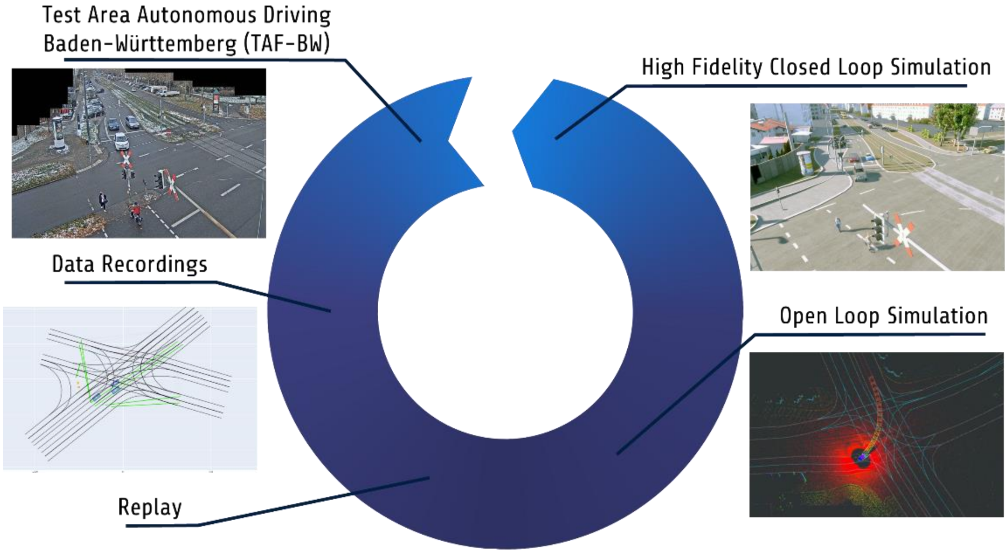


Figure 34: Concept for testing highly automated vehicles in different test stages within a test area

3.3.13 Proving ground – harsh weather performance (Mercedes)

https://www.vvm-projekt.de/fileadmin/user_upload/Final_Event/Poster/VVM_FE_Poster_14.09_MBAG_Schlechtwetterperformance_v2_web.pdf

3.3.14 Proving ground – advanced pedestrian dummy (Mercedes)

Increasing need for real world testing

The development of driver assistance systems has made a drastic leap forward, with the clear goal of reaching higher automation levels and ultimately autonomous driving in the future. One focus of development is the improvement of road user detection as well as early intention recognition based on their movements. Nevertheless, system testing has been limited to “static” VRU dummies with purely two-dimensional translational degrees of freedom. To address this gap, the approach taken was as follows:

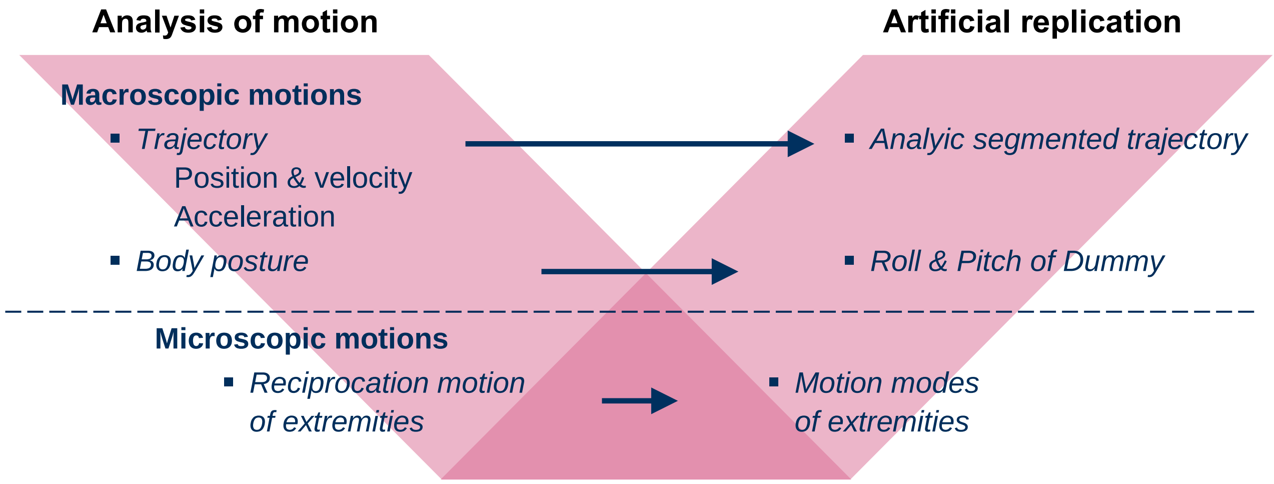


Figure 35: Abstraction model of the motion patterns

Measurement of macroscopic movements of pedestrians and cyclists

To generate a data basis, a subject study was carried out. Both, pedestrians and cyclists, were equipped with measurement systems consisting of a rtk-GNSS system and an inertial measurement unit (IMU), as can be seen in Figure 36. The following movements were determined:

Trajectory		
Velocity	✓	✓
Corner velocity	✓	✓
Acceleration	✓	✓
Deceleration	✓	✓
Posture		
Roll angle (curve)	✓	✓
Pitch angle (acc.)	✓	
Pitch velocity (acc.)	✓	

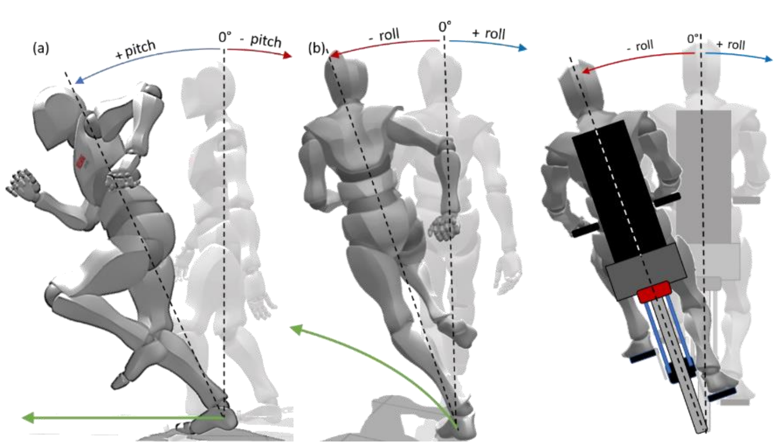


Figure 36: Examined macroscopic values for pedestrians and cyclists

Analysis and artificial replication of selected macroscopic motion patterns

Whereas typical velocities for walking, jogging and running are widely available in scientific literature, distribution of pitch and roll angle and their progression over time was subject to this study.

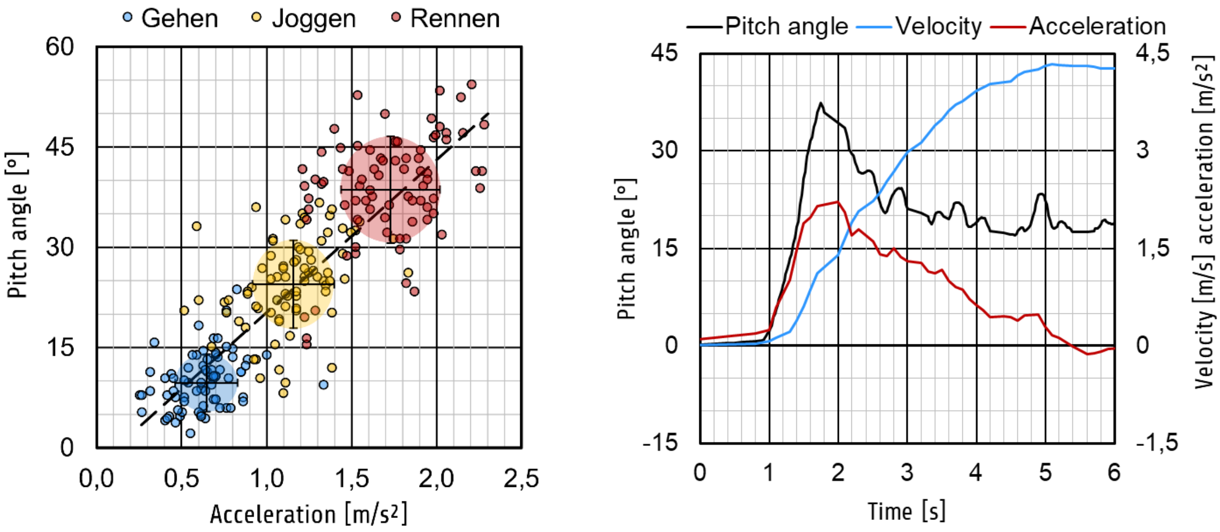


Figure 37: Distribution and of peak pitch angles during acceleration for different gaits (left) and progression of pitch angle, velocity and acceleration during start phase

As shown in Figure 37 (left), peak pitch angles during the start phase show a clear dependency regarding acceleration, which therefore was described analytically. The temporal course (Figure 37, right) also shows a positive pitch angle (leaning forward) at constant velocity.

Measurement of microscopic movements

Movements of extremities were measured by a Motion Capture System consisting of ten IMU sensors. Placement of the sensors and the derived microscopic motion values are shown in.

- Step rate
- Hip angle (over time)
- Knee angle (over time)
- Elbow angle (over time)
- Shoulder angle (over time)

Distinction between gaits

Behaviour at velocity/gait change

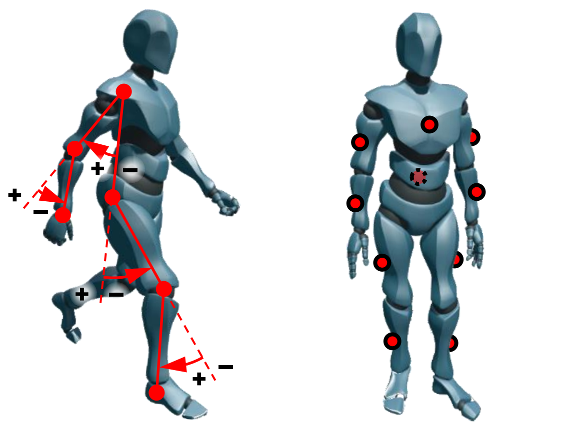


Figure 38: Examined microscopic motion values (left, middle) and placement of motion capture sensors

Analysis and artificial replication of selected microscopic motion patterns

As shown in Figure 39, knee and hip angles differ strongly between subject and standard NCAP dummies. Especially for high velocity gaits, the articulation of the dummy is not capable of providing enough force to control the angles precisely. Since the knee joint isn't controlled actively, the highest deviations can be seen here. A clear improvement can be reached by using advanced actuation methods as depicted for the advanced motion dummy (Astero).

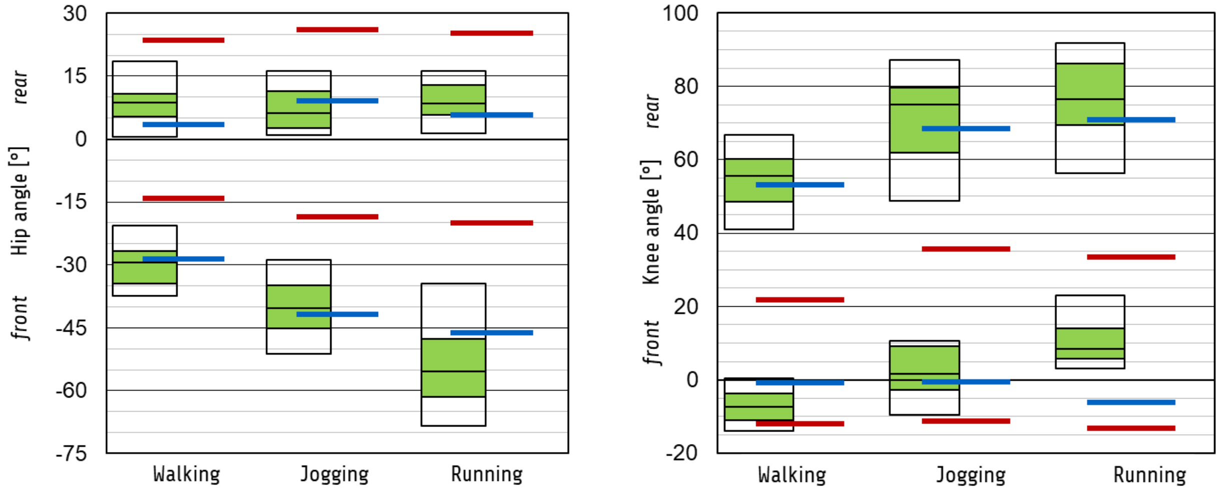


Figure 39: Comparison of front and rear maximum hip and knee angle for subjects, standard NCAP dummy and advanced motion dummy at different gaits

Conclusion

Using this procedure, a prototypical implementation was realized (Figure 6) to reach a more realistic motion behaviour for VRUs. Thus, Mercedes-Benz is able to include intention detection using slight indicators like shift of centre of gravity or hip and knee angles. This enables new perception strategies, leading to a higher level of real life safety.

3.3.15 Proving ground – scenario based testing (BMW / Akka)

Main Goals

Automated execution of complex test scenarios (e.g. driving dynamics, endurance test) on a testing site to ensure reproducible results in a safe environment.

Seamless interaction of proving ground tests with simulation to ensure evidence-based results.

Technical Goals

Verification of Trajectory planning of the ego vehicle for simulation and real testing on a common standard as defined in the scenario.

Development of a control center as central unit coordinating all automated test participants (vehicles, moving platforms, e.g.)

Safety

Geofencing surveillance of proving ground (managed classical / autonomous traffic)

Mission

Implementation for VVM traffic scenarios with coordinated robotics incl. examplery demonstration of the functional use case (FUC2.3).

Scenario scenery description based on the definition of the ASAM OpenScenario 2.0 and OpenDrive1.X standard

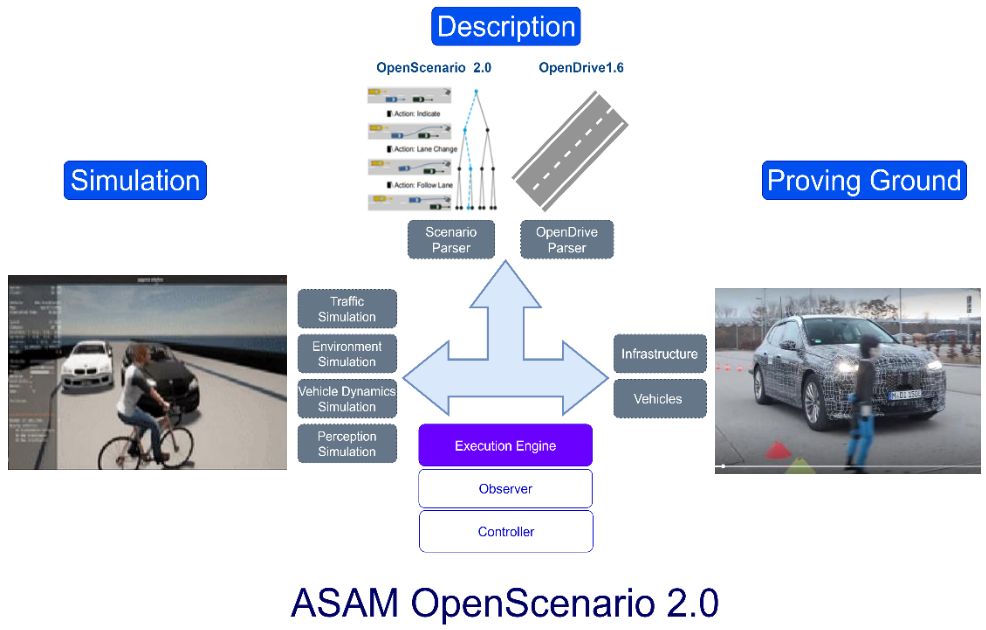


Figure 40: Open Scenario as a common format for scenario description in simulation and for real testing

Possibility to include different test platforms with different interfaces in one test set. Reproducable control of traffic participants and dynamic road infrastructure

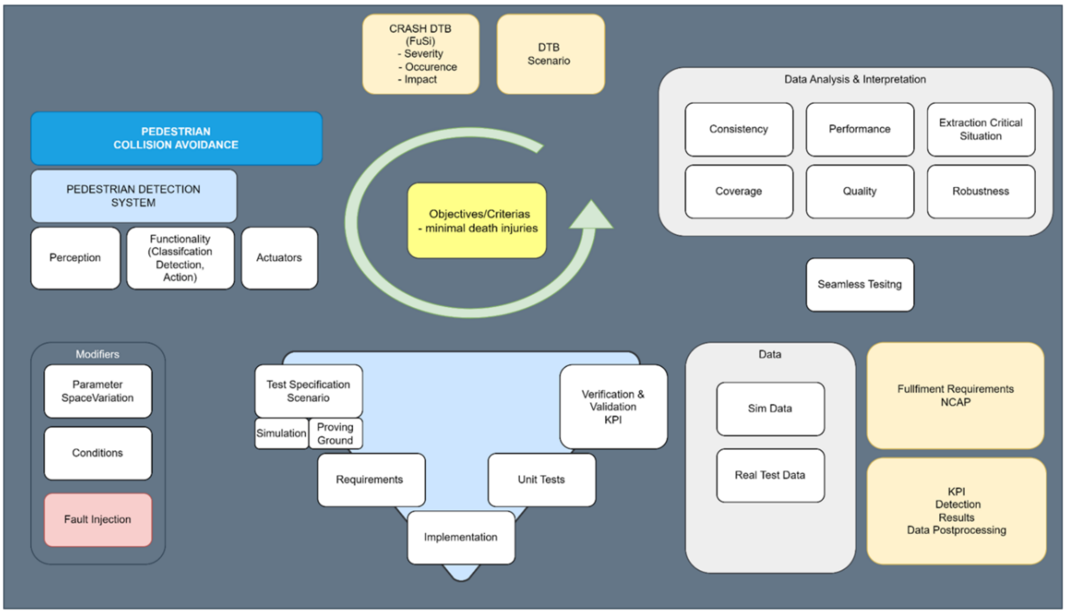


Figure 41: Scopes of use and application for an automated test environment / automated proving grounds

Realization

Vehicle control with use of robotics and/or ECU manipulation

HMI interaction with actuators/robotics (simulation of driver activities)

Development a ROS2 framework controlling an automated multi-participant test (vehicles, moving platforms) via the control center

Integration of a moving platform/target into the ROS2 framework using the example of 4activeSystems' Freeboard 4activeFB-small (ROS2 Interface, Trajectory planning of moving platform, Control and surveillance of moving platform, Remote manual control from control center with use of joystick)

Trajectory generation (for all test participants) based on ASAM OSC2.0 scenario description

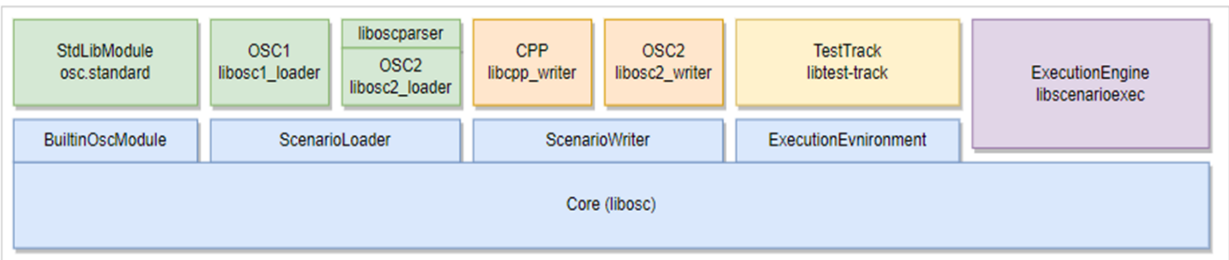


Figure 42: Development components for trajectory extraction based on the ASAM OSC2.0 standard

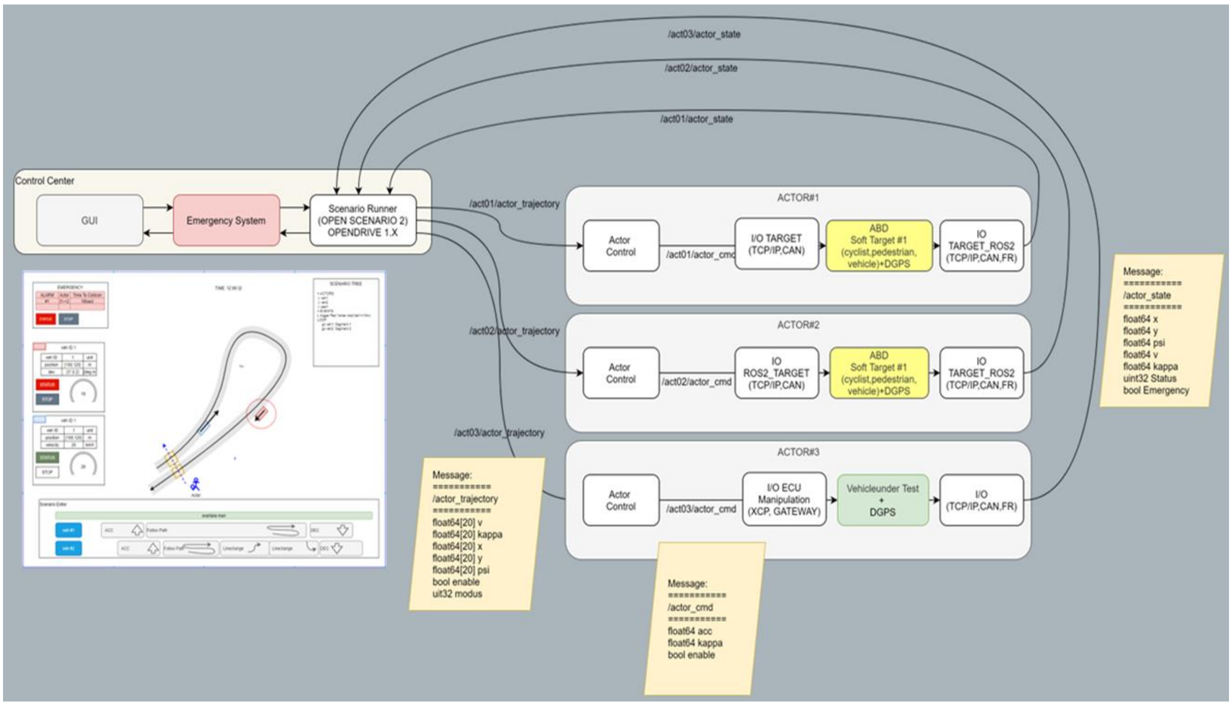


Figure 43: ROS2 framework for integration a lot of test participants in a test scenario

3.3.16 Fleet Monitoring & Assessment (FMA)

Main Goals

The V&V concept must provide sufficient test coverage while ensuring feasibility and the highest possible efficiency to provide adequate evidence in accordance with the safety argumentation.

System validation ensures that the system fulfills its intended purpose and thus meets the needs of the stakeholders for the system. The essential aspect here is society's expectation of the safety of the system.

FMA is a real-time monitoring approach that efficiently supports the validation of an AD system throughout the product lifecycle, both in Field Operational Tests (FOT) for system release and during serial operations to maintain the safety of the system in an "open traffic context".

FMA supports to provide the evidence that

- the residual risk due to unknown unsafe events and system emergences is acceptable low,
- known uncertainties do not affect the safety of the system.

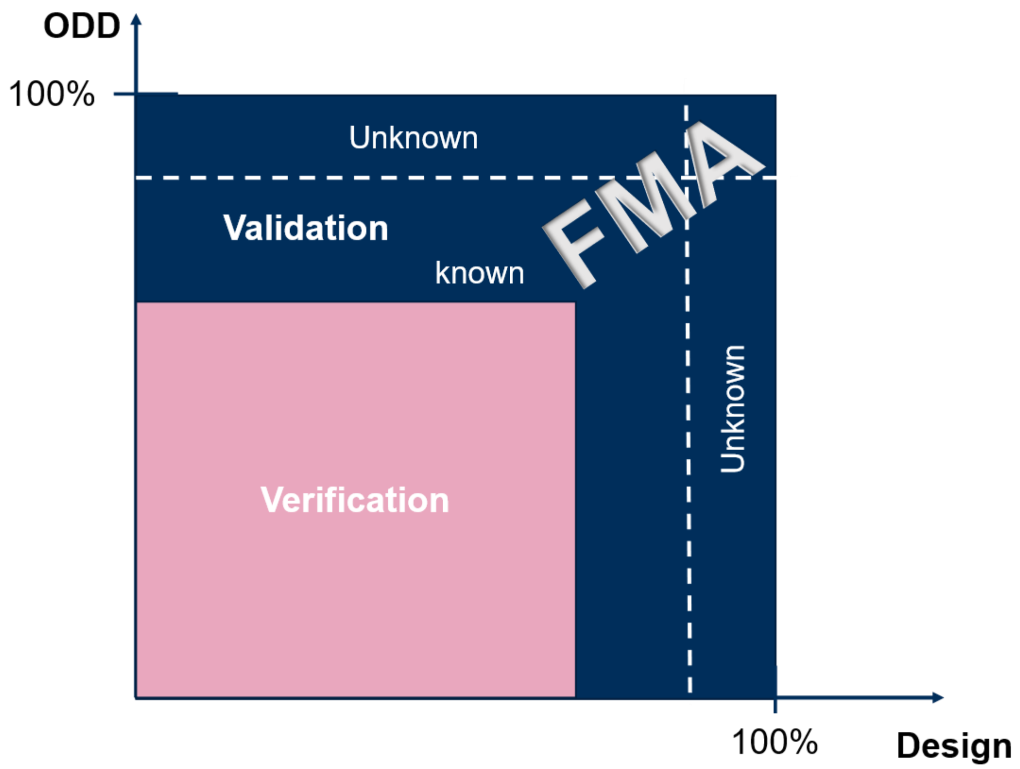


Figure 44: Verification and Validation aspects

Realization

The FMA System provides a framework for monitoring and evaluating the performance of an HAD vehicle during operation.

The FMA System

- provides all the data required to evaluate functionality,
- enables event recording and data storage,
- enables the transfer of stored data to a backend/cloud for analysis,
- provides an interface that allows an operator to configure the system for a specific campaign.

To reduce the amount of data and analysis effort, data is only collected in scenarios that show unexpected or unknown traffic behavior. The expected behavior can be defined in a campaign and implemented as a trigger.

Basic concepts for FMA

There are various conceivable approaches to applying FMA to automated driving, which pursue different objectives and differ from one another in terms of the scope of the system and its implementation requirements. These approaches are fundamentally independent of whether they are used before or after the start of series production.

1. provision of data:
   Recording of sensor (raw) data for re-simulation (open loop)

2. test of perception:
   Check of anomalies and inconsistencies between sensors with different designs or technologies (open / closed loop)

3. test of automated driving functions:
   Identification of differences when comparing the planning and specifications of the passive automated system with the observed behavior of the driver (open loop)

4. test of the automated system behavior:
   Monitoring of critical phenomena in interaction with other road users and the traffic infrastructure (closed loop)

Exemplary implementation to demonstrate the FMA system

The demonstration concept consists of two parts:

1. presentation and demonstration of algorithms that process field monitoring data to recognize defined conditions (trigger logic) and then extract this data when these defined conditions are met

2. presentation of the entire FMA tool chain (triggering - data acquisition - data storage - data transmission - data analysis) in a Continental test vehicle

Re 1) Presentation and demonstration of algorithms that process FM data.

As part of the final presentation, trigger logics are presented and demonstrated using the example of the FUC2.3 which pursue the following test objectives:

1. monitoring pedestrian detection during operation

2. testing the system behaviour in interaction with pedestrians

3. detection of previously unknown critical situations and unsafe events involving pedestrians

4. statistical statements on remaining unknown scenarios to assess the residual risk when testing scenarios with pedestrians.

Figure 45: VVM Functional Use Case (FUC) 2.3

The following was demonstrated:

1. Rule-based triggers

   - Pedestrian detection and tracking: Detection of the presence of pedestrians in the defined ROI (region of interest)
   - Trigger on sudden appearance of pedestrians from occlusion in the defined region of interest

2. Data-driven trigger:

With the data-driven triggers, the aim is to identify previously unknown critical phenomena in interaction with other road users and the traffic infrastructure.

Two AI-based approaches are presented in the final presentation.

- "Event detection" approach

  This is an AI function that recognizes events/scenarios in a sequence of input images and assigns them to "known" categories as they usually occur during vehicle operation, such as normal driving, sudden braking, stopping, turning, lane changing, etc. Events/scenarios that cannot be assigned to any of these "known" categories are categorized as "anomalies" and trigger measurement for data acquisition. Anomalies can be, for example, the camera falling. The recorded measurement data can then be analyzed offline in a backend to assess whether these are previously unknown and possibly safety-critical events/scenarios that require system improvements/changes. Furthermore, new events/scenario categories may arise.

  However, measurement data acquisition can also be triggered by any "known" category if it is considered potentially safety-critical, such as a sudden high deceleration (sudden braking).

- Generalized Event Discovery approach (GED)

In addition to assigning unlabelled data to a category, the GED approach also makes it possible to generate new categories in FMA data analysis. These new categories can be evaluated in further analyses regarding their criticality and triggers can be developed on this basis. The triggers can then be used for self-learning systems as part of a safety management / DevOps process.

The following figures show the inference pipeline for the "Event Detection" and "Generalized Event Discovery" approach.

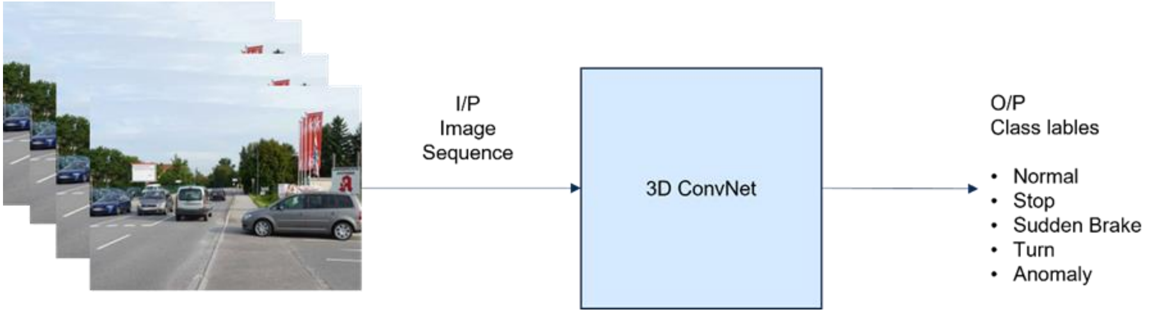


Figure 46: Interference Pipeline - Event Detection

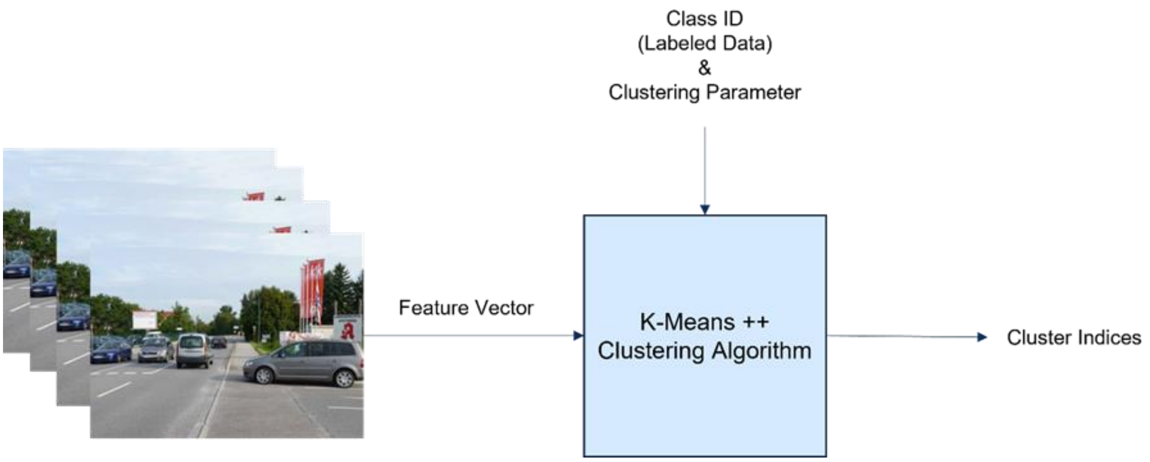


Figure 47: Interference Pipeline - Generalized Event Discovery

To evaluate the "Event Detection" approach, the following event/scenario categories the following examples were defined and specified using GPS position, yaw rate and longitudinal and lateral acceleration.

- Normal driving
- Sudden braking
- Stopping
- Turning

For data preparation, the classes are defined via user-defined threshold values using the above-mentioned measured variables.

In the following figures, the red line shows the user-defined threshold value for selected data set from the BDDK100 data base (https://www.vis.xyz/bdd100k/) for "Stopping" and "Sudden braking" as an example

Stopping: Speed V < 2 m/s for longer than 5 seconds

Sudden braking: deceleration a < 0.4g

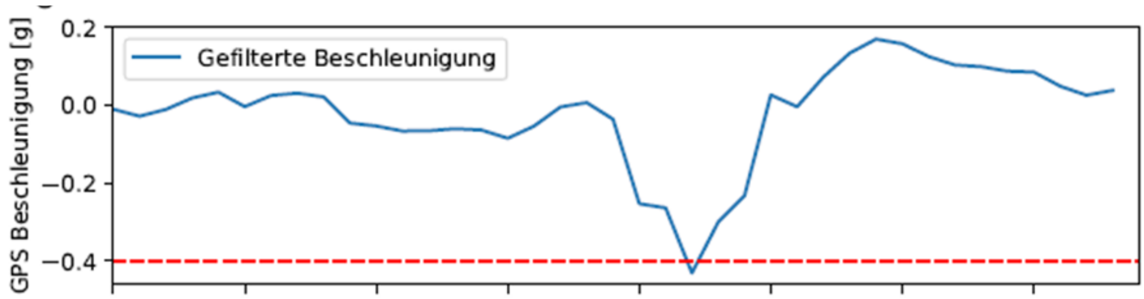


Figure 48: Data preparation for event classification - Stopping

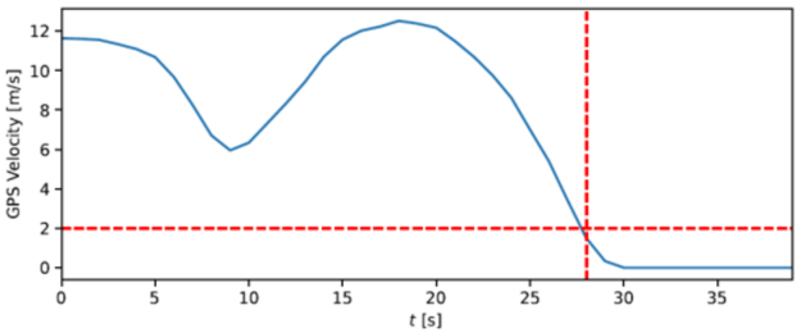


Figure 49: Data preparation for event classification - Sudden Brake

An important observation is that these user-defined threshold parameters are dataset-dependent. Therefore, these parameters must be determined on a dataset-specific basis after an appropriate statistical analysis.

The following figure shows an example of images from video sequences of the specified categories

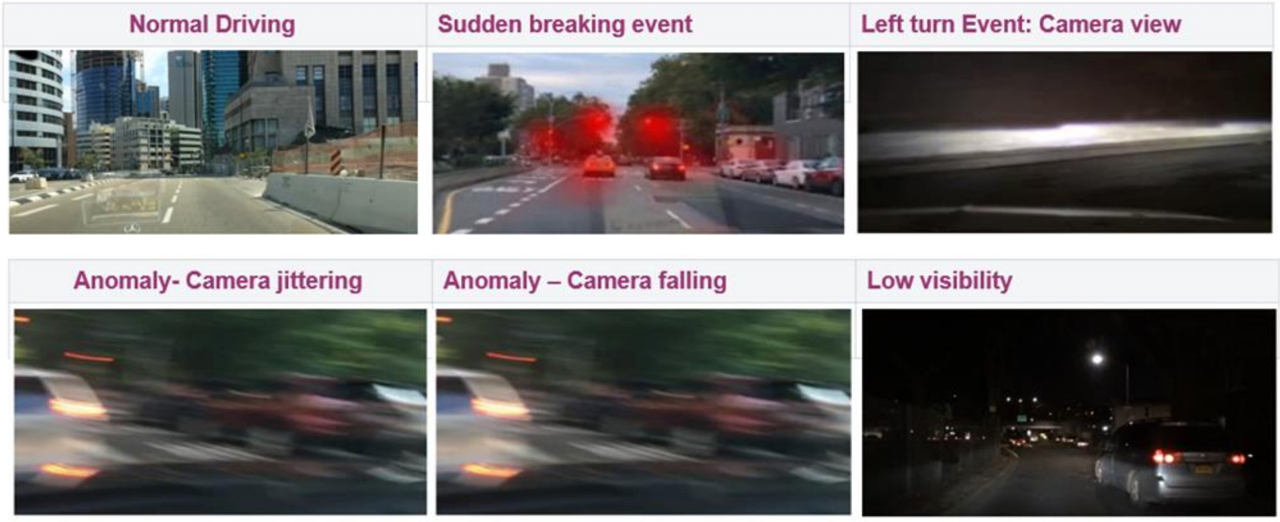


Figure 50:Example images of the different event/scenario categories

Subsequently, video sequences from a BDD100K data set and a Continental data set were labeled using the corresponding synchronized measurement data (GPS position, yaw rate and longitudinal and lateral acceleration) and a 3D convolutional neural network was trained with this video data.

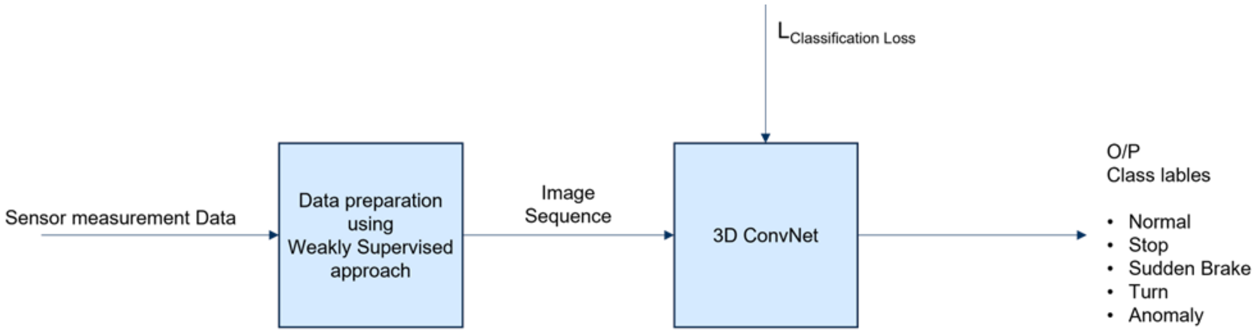


Figure 51: Training Event Detection

Finally, the performance of the "Event Detection" was tested with appropriately labeled video sequences from the BDD100K data set and Continental data set. Python tools were used to analyze the results. The following figures show the evaluation of the results in a confusion matrix.

In total, the trained network was tested with approx. 14000 (BDD100K) or 1500 (Continental) video sequences. The accuracy was 79.85 % for the BDD data and 77.79 % for the Continental data.

BDD Data

True Label \ Predicted Label	Normal	Stop	Sudden Brake	Turn	Anomaly
Normal	3467	282	260	45	545
Stop	594	6120	547	103	106
Sudden Brake	31	42	396	6	21
Turn	11	9	13	1200	50
Anomaly	114	17	45	56	287

Continental Data

True Label \ Predicted Label	Normal	Stop	Sudden Brake	Turn	Anomaly
Normal	919	26	75	22	131
Stop	7	174	6	35	1
Sudden Brake	17	13	19	7	6
Turn	0	1	0	149	13
Anomaly	0	0	0	0	0

Figure 52: Confusion matrix – “Event detection” result with test data

The two diagrams above are dominated by the diagonal elements, which show the overall correct classification of event categories. The other elements in the table show a different classification to the actual "labels". The element in the upper right corner shows that approximately 10% of the data labelled as "Normal Driving" was categorized as an anomaly.

There were no samples of the anomaly class in the Continental data, so the bottom row of the right-hand table is empty.

Re 2) Representation of the entire ST tool chain (triggering - data acquisition - data storage - data transmission - data analysis) in a Continental test vehicle

To present and demonstrate the tool chain, an FMA concept was implemented in a Continental test vehicle to detect critical situations involving pedestrians. The aim was to integrate a trigger function with computer vision-based AI functionalities for pedestrian detection and to demonstrate this in a use case (FUC-2.3).

A measurement is triggered when a pedestrian is detected at a very short distance from the vehicle in the field of view (FoV) of the rear-facing camera integrated in the side mirror. This example is used to demonstrate that the vehicle has passed a pedestrian even though he is standing directly at a crosswalk or has already entered it and a critical situation has therefore arisen.

The trigger definition is based on an existing object detection pipeline at Continental and uses the 2D coordinates of the object list through semantic segmentation. The trigger is activated, and measurement data collection is triggered as soon as the pedestrian is recognized through semantic segmentation in a defined ROI (region of interest). The ROI represents a defined area around the vehicle.

The following figure shows the data flow pipeline.

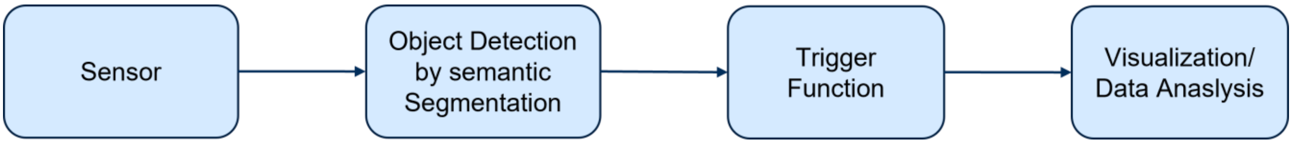


Figure 53: Data Flow Pipeline

The following figures show an example with active and inactive triggering. The camera images show the field of view to the rear to the right and left of the vehicle. If a pedestrian is detected through semantic segmentation in a defined ROI (region of interest) (white bordered rectangle), a measurement is triggered, which is indicated by a colored (red) frame.

	Region Of Interest
	Trigger Active – near miss with a pedestrian
	Trigger Active – no near miss with a pedestrian

Figure 54. Legend – Visualization

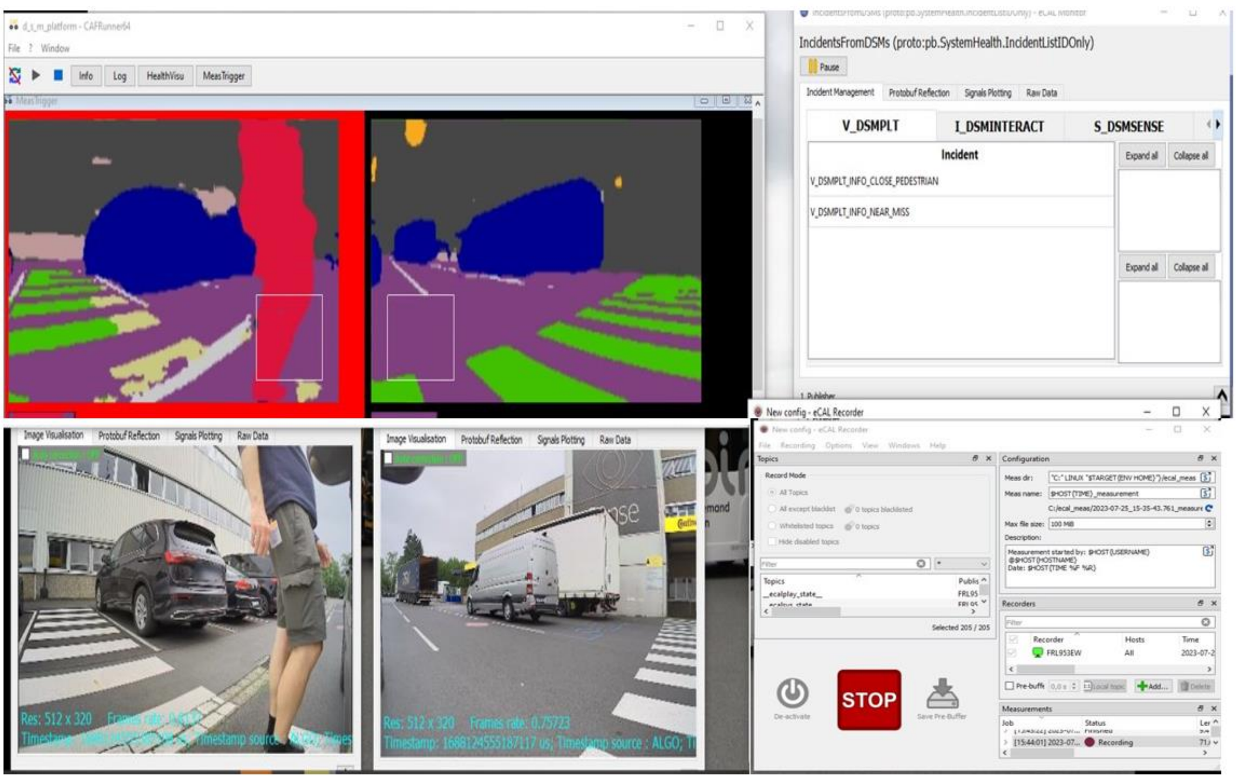


Figure 55: Example – Trigger active, near miss with pedestrian.

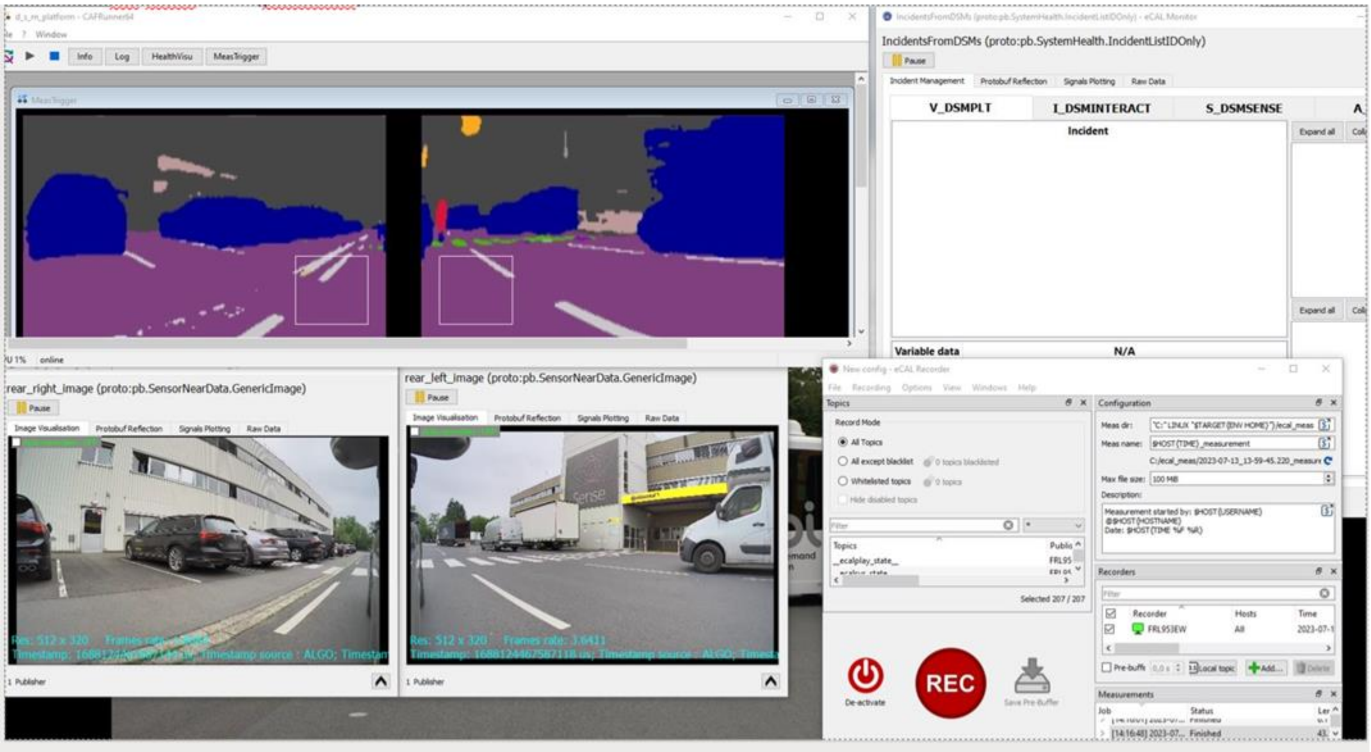


Figure 56: Example - Trigger Inactive, no near miss with pedestrian.

Using the example of a wired data transmission from Vehicle storage to an external PC via Vehicle Network was used to demonstrate, the data transmission from the vehicle into cloud/ backend was demonstrated as an example.

A Continental evaluation tool was then be used to demonstrate a data analysis on the PC, e.g., as it can be carried out in a backend in real operation. The following figure shows an example of a map section with the driving rout and the position of critical events/ situations.

Figure 57: Map section with the driving rout and the position of critical events/ situations.

3.3.17 Proposal on “Separation of concerns, identification of cross-cutting concerns” (Bosch)

Target of the approach of “separation of concerns” is to systematically deduce the test space which has been removed by the decomposition of the system (functional and technical architecture). The removal of the test-space-areas arise because of the reduction of interexchange of subsystems. Thus, the advantage of decomposition of the system and the related decomposition of test for subsystems create uncertainties coming up by not tested emergences within the system. Those system-emergences may lead to risk. Thus, the approach of “separation of concerns” deduces risk relevant test spaces in order to mitigate risk.

Premises for technical setup of separation of concerns is a functional decomposition and a functional separation of concerns.

Rough process steps for functional decomposition and a functional separation of concerns.

Functional decomposition according to the functional architecture.

Assignment of scenic functional challenges to functional limit performance of the functional elements. e.g. by formal analysis of functional limits on the basis of argumentation requirements and validation requirements via test objectives and test methodology as well as the safety model.

Determine the “coupling” of functional elements and deduce the effect (e.g. by numerical simulation).

Assign scenic challenges to the “coupling” areas.

When applying a functional separation-of-concerns (here the industry-standard division of the functional architecture into "Perception - Planning - Actuation"), a functional emergence basically results from the technical implementation of the function in a vehicle. Functional emergence here refers to systemic effects that can be described in functional terms, which only result from the technical implementation.

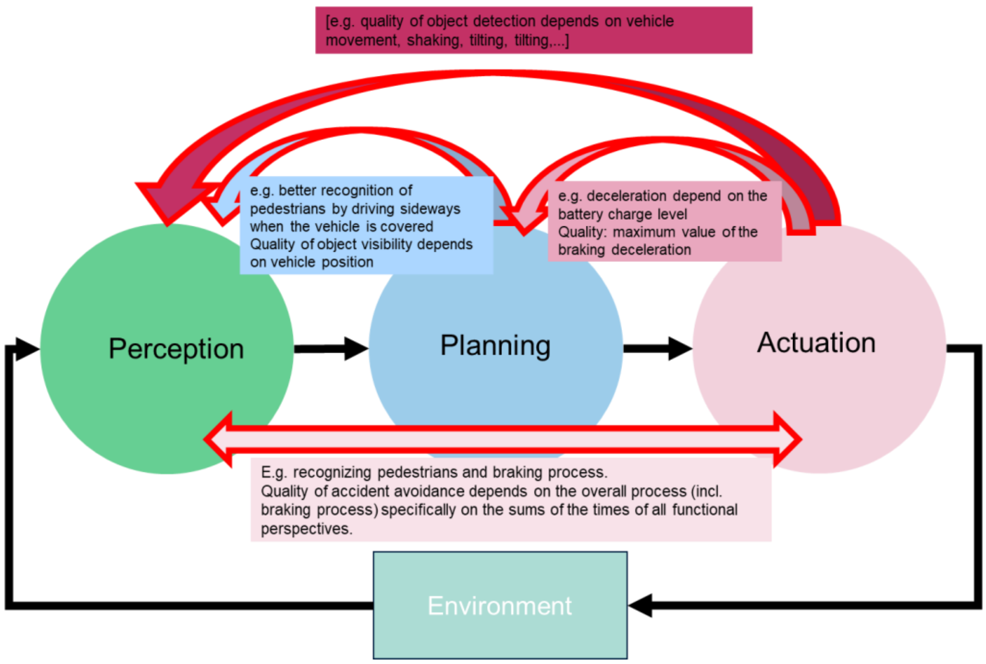


Figure 58: Example of functional decomposition of in automated vehicles and related emergences

The figure above illustrates that elements of a functional decomposition (separation-of-concerns) are linked by properties of the implementation, namely via interactions not shown at the functional level. With regard to the goods at the level of the vehicle ("as a product" or also "in traffic"), this results in cross-cutting concerns between the tasks assigned to the various parties (organizational lines, suppliers). Finding functional emergence from cross-cutting concerns is the subject of functional emergence analysis.

The functional emergence analysis is carried out through reciprocal steps of analysis and synthesis of technical target variables. The results of the analysis then flow into the determination of scenic-functional boundary areas.

The proof of the safety of functional emergence is part of the release argumentation. Conversely, requirements for testing in boundary areas result deductively from the argumentation method. This deduction takes place, for example, based on the logical and technical architecture and the emergence analysis based on it. This reciprocal relationship between analysis and argumentation is not a contradiction but is characteristic of directed technological developments.

The results of the functional emergence analysis can then be used to determine scenic-functional limit ranges, which creates an analogy to the usual testing of systems with limit values of the inputs of functional elements.

The functional emergence analysis thus fits into the analytical scenario-based test planning using a test space matrix and makes a decisive contribution to test coverage, especially when safeguarding technical performance.

At the goal of a directed technological development, i.e. when technical maturity is reached, the method can be further used to evaluate and plan intended technical changes.

The results of the functional emergence analysis can also be used to determine the need for V&V activities in the event of changes to the CoU (context of use), i.e. planned changes to the operational design domain (ODD) or planned or observed changes to the operational domain (OD).

The functional emergence analysis methodology can be used to determine those areas in the operation of technical elements where there is a particular need for testing (functional limit performance).

Decomposition of the technical system into domain-dependent perspectives for test/testing automotive

Decomposition of the technical system into (automotive domain-dependent) perspectives suitable for testing/inspection (typically strongly based on logical architecture) is a necessary precondition in order to separate concerns. The decomposition into Function, Distribution and Component is an approach to cover established automotive domain and interface structure.

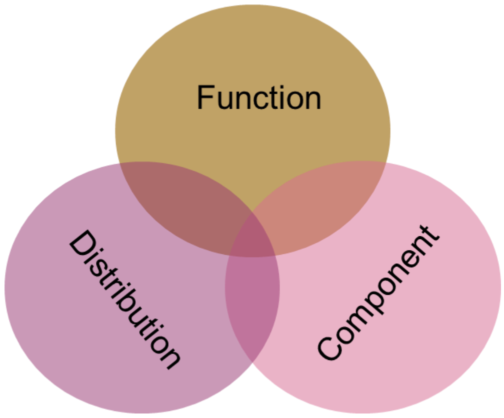


Figure 59: Decomposition approach of the technical system according to automotive domains

It is of course possible to decompose in different ways. How ever - the advantage of this decomposition approach results in the ability to later on divide and distribute test-efforts. As an example, for this advantage, we can state that if the functional performance of a component is specified, and its interfaces fully cover the specification as also all specifications towards distribution (network) are specified then we can define tests for the “component only” which only dependent on this the specified interface. Also, some of the tests of the component will be independent or marginal dependent of the interface. Thus, tests and its results may be remained valid and useful for complete V&V process even if some specification will change. This enables to define tests where system-environmental conditions can be assumed as fixed as for example power supply or thermal exchange. For example, topological analysis according to IEC60812 (FMECA) can be used for the analysis.

One assumption for this decomposition approach is that all behavior-relevant functions are fully mapped in the functional architecture.

Determine the coupling of functional elements with technical properties and deduce the effect

After the identification of critical qualities through the analysis step of the functional emergence analysis, a synthesis step takes place in which functional and technical target values are determined and optimized with regards to the feasibility of tests.

Process Steps Level1 : Allocation of Function on Component and or Distribution - deriving the Technical Test-Specification (1. level)

process of iteration

Assignment of functional target performance (by e.g. (Re)-Analysis of functional limits (functional specification from design) to technical design.

Assignment of technical target performance (technical specification from technical design)

Assignment of scenic functional challenges to functional performance limits of the technical elements.

Assignment of physical system and test- infrastructure (test-instances)

Transform or decompose or compose until functional performance can be expressed by technical performance. (Physical application and test distribution to test-instances is to be done in a next step – not focus here).

Remark: Consideration of non-functional performance (based on e.g. logical architecture) also assign further separation of perspectives e.g. on energy supply or energy supply to distribution.

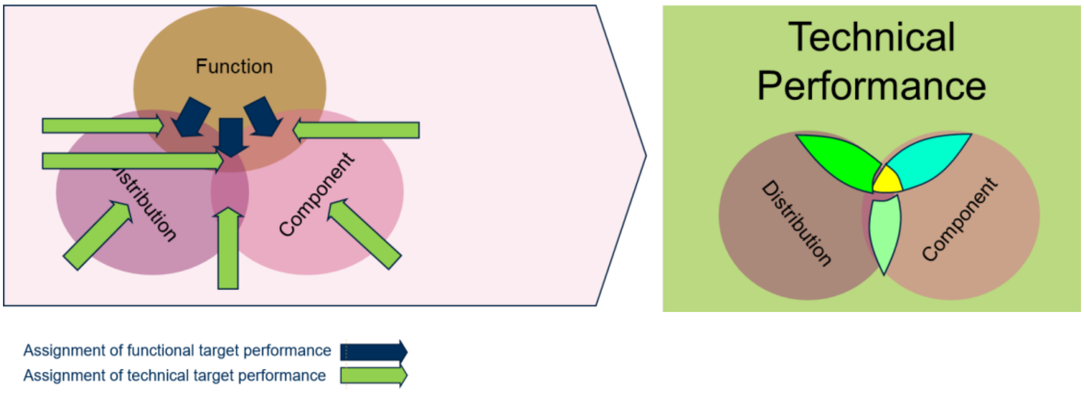


Figure 60: Assignment of the nominal technical performance aligned with the functional performance.

Examples of fields for analysis

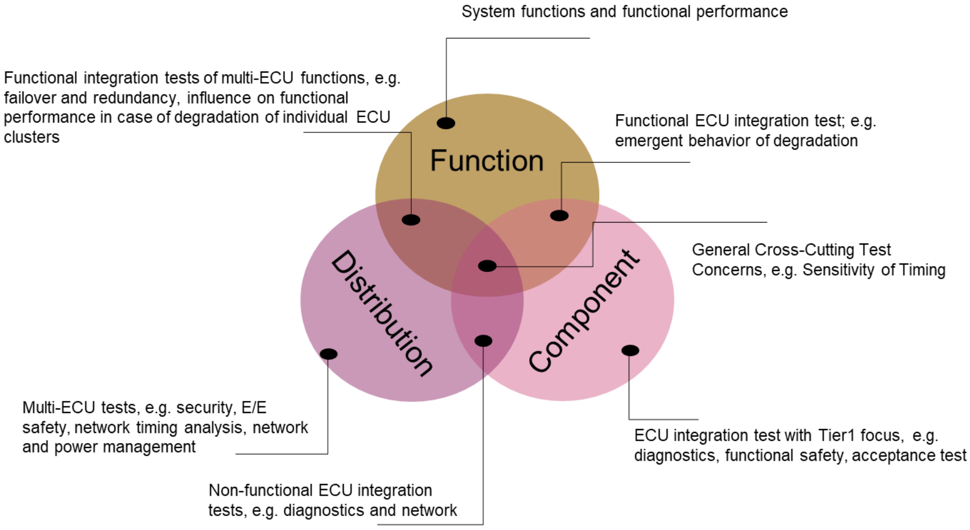


Figure 61: Examples of for assignment of nominal performance tests to the decomposition structure of automotive domain

The result of the first step of assignment is a matrix in which the technical / physical elements of the system (columns) are aligned with the scenic challenges including adverse conditions (e.g. weather) and the test-infrastructure /test-instance (rows) and where the matrix elements contain the technical performance to be measured**.**

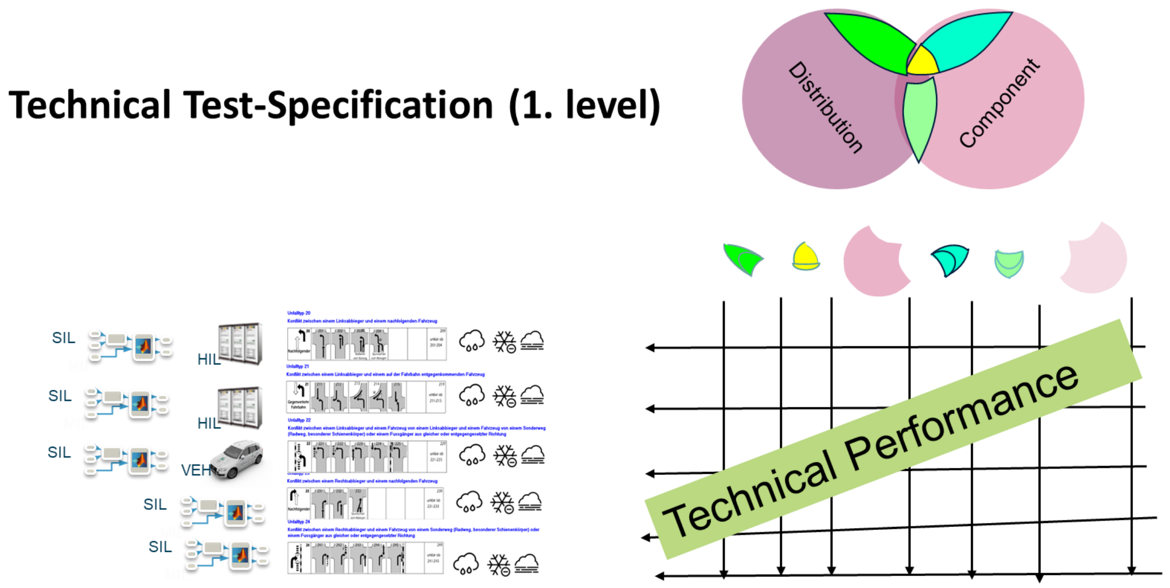


Figure 62: Graphical illustration of the Technical Test-Specification (1. level)

Process Steps Level2 : Systematical technical emergence analysis - Separation of concerns at technical level - deriving the Technical Test-Specification (2. level)

Systematic coverage of analyses of emergences / couplings of system perspectives via impact path analyses. Possible are additional design checks e.g. check whether the implemented performance of the component fulfills the expectation of the nominal performance in the function. This is based on the specific scenarios via impact path analyses (formal methods: FMEA, probFMEA/CFT, DRBFM,…).

Process of iteration

Use of the logical architecture to identify concerns arising from non-functional systemic couplings, e.g. power supply, bus communication

Derivation of non-functional (e.g. topological) performance requirements for components and distribution (orthogonalization, free-cutting)

Inclusion of technical concerns

Assignment of test platform and scenarios according to functional Test-Specification and evaluated (concern) scenarios.

Both aspects of verification (nominal performance) and validation (validity of the specification of technical properties) must be determined here. The specified and expected production variance places high demands on the planning of the testing of technical elements in the product development process (automotive suppliers control the latter through production-related QM systems).

Operation-dependent properties (e.g. ageing or soiling of sensors) belong, similar to the completeness of the specification, to the area of validation

Example for Process Steps Level2: Analysis of the Nominal Performance for Localization

(Check whether the implemented performance of the component fulfills the expectation of the nominal performance in the function.)

The example shows many impact path analysis fields and related qualities for test definition for the functional example of perception by localization.

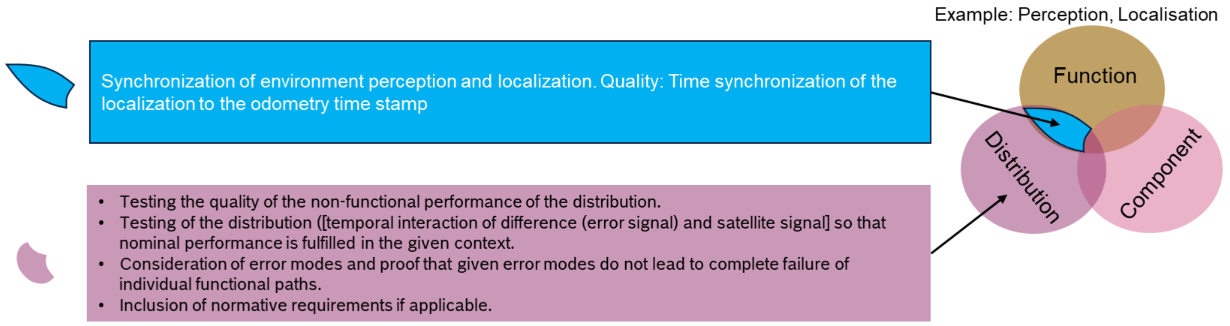


Figure 63: Example for Localization at process step2 (Function vs Distribution)

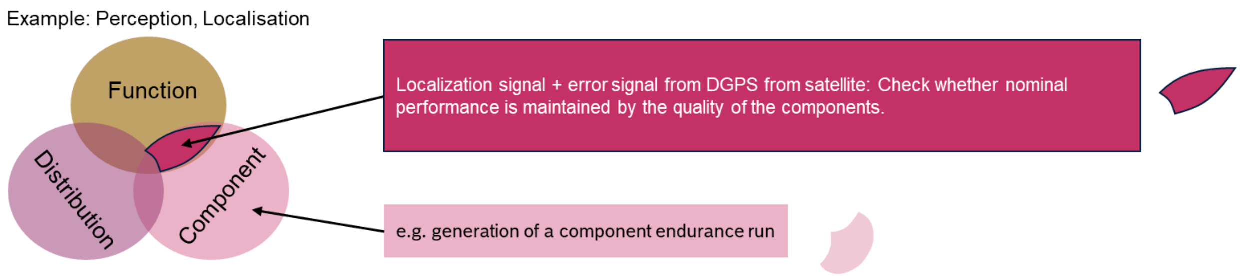


Figure 64: Example for Localization at process step2 (Function vs Component)

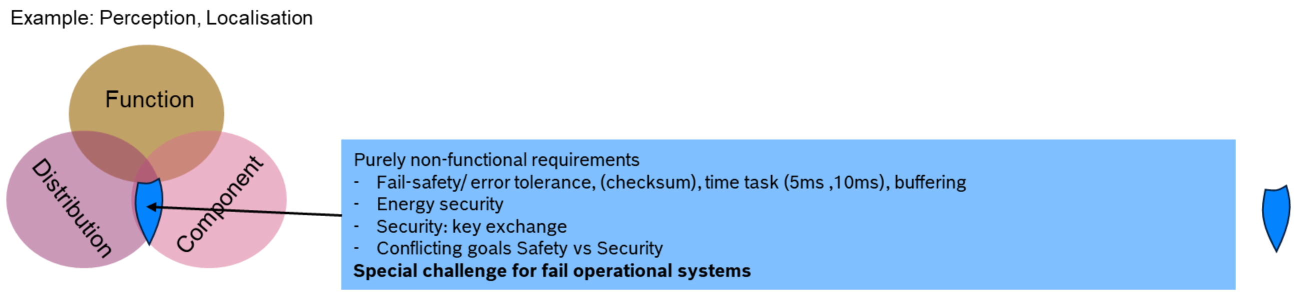


Figure 65: Example for Localization at process step2 (Distribution vs Distribution)

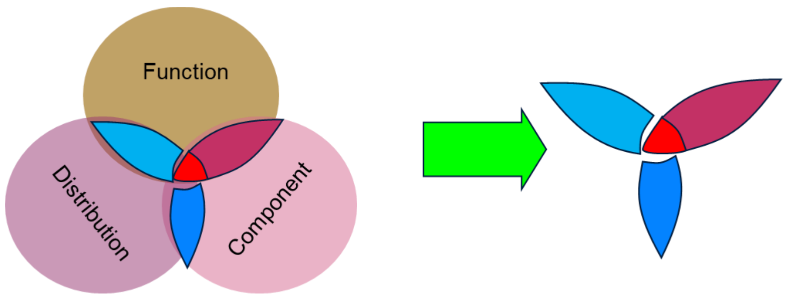


Figure 66 graphical illustration of the result of the impact chain analysis according to chosen decomposition, highlighted colors show fields of emergences

The results of the second step are the additional test resulting from the emergence analysis.

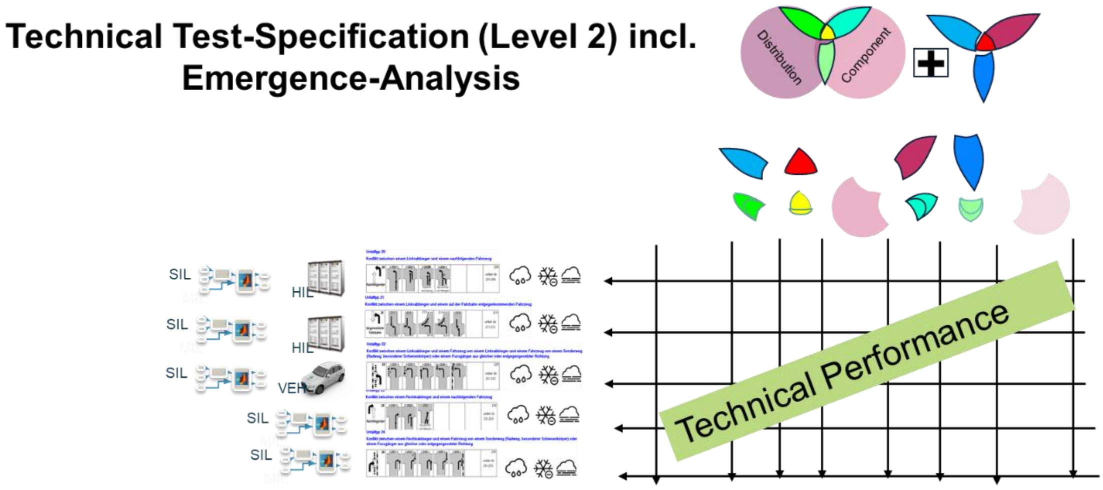


Figure 67: Graphical illustration of the Technical Test-Specification (2. level), the technical performance resulting from emergence analysis is added to the columns.

In the automotive industry, testing is a technology with several different disciplines, which are used either for testing aspects of a complete real vehicle or for isolated supplied components. In addition to testing on real objects, simulation is also used.

According to the system of representation of the system-of-interest (SoI) with regard to a test requirement as a system of "tester" and "implementation-under-test" (IuT) with points-of-control-and-observation (PCO) at the interfaces, which was borrowed from IEC9646 in one of the previous reporting periods, the tester and IuT can be related as shown in the following diagram:

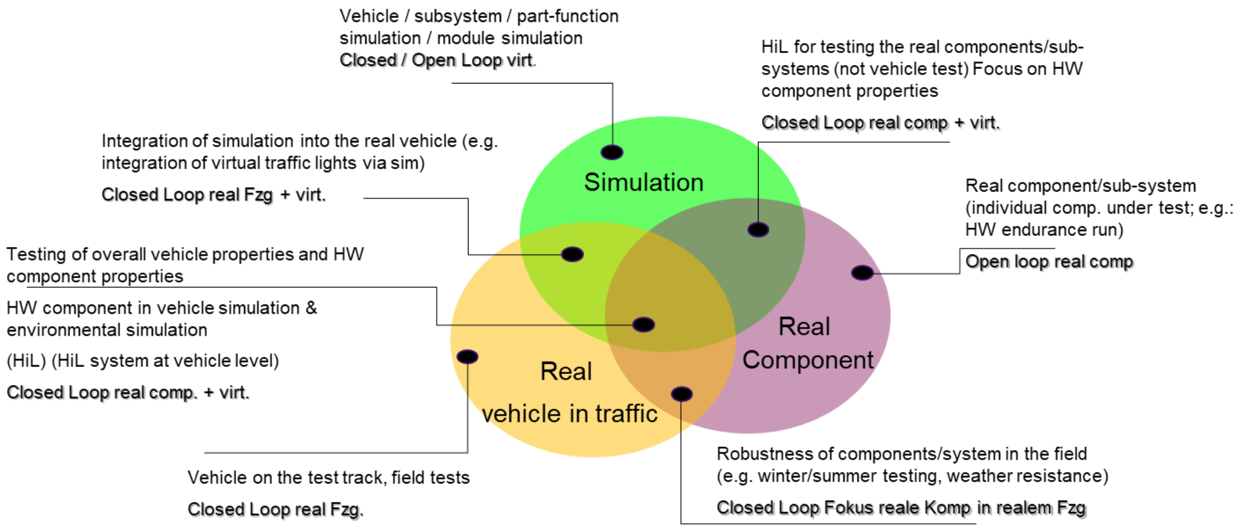


Figure 68: Systematics of vehicle test systems

The selection of suitable methods is determined by the question of interest (QoI) as well as the effectiveness and cost of test solutions. The production of a large number of real vehicles, for example, is associated with considerable costs, especially if they have to be equipped with special instrumentation for testing complicated real components. The costs for this are many times higher than for near-series vehicles, which are equipped with environmental sensors and computer technology for highly automated driving, for example, but where the testing of vehicle technology aspects is in the background.

Aspects of testing highly automated vehicles using simulation are fundamentally limited to assumptions about the vehicle's environment and the theories of interaction with the vehicle based on this. Test planning must take this into account when selecting real-life testing and simulation in order to ensure the statistically sound support required for the release argumentation.

For vehicle-related model properties, this can be done, for example, by model validation in the real vehicle (comparative measurements). The open world around the vehicle-related models can be achieved, for example, by an analytical determination of monitors that are operated in the "vehicle in traffic" system.

4 Conclusion

The Test Orchestration follows the concept of the division into functional and technical level and distributes concrete test cases to test instances. Based on the Test Orchestration, the test execution realizes seamless testing to efficiently provide test data. Finally, test evaluation is done based on the provided test date. The challenge of aggregating multiple results on the functional level can be mastered by using a parameterized risk model.