Chapter

Virtual Verification and Validation of Autonomous Vehicles: Toolchain and Workflow

Alexandru Forrai, Mohsen Alirezaei, Tajinder Singh, Amit Gali and Jeroen Ploeg

Abstract

The complexity and efficient development of Autonomous Vehicles (AVs) require robust testing and validation methods to ensure their safety and reliability. This book chapter presents Siemens' autonomy toolchain for testing and virtual validation of Advanced Driver Assistance Systems (ADAS) and AV functions. The approach integrates scenario-based testing, virtual validation, and compliance with the applicable regulations and standards, such as the International Organization for Standardization (ISO). Relying on the multi-pillar safety validation framework proposed by United Nations Economic Commission for Europe (UN-ECE), and EU regulations, Siemens' toolchain enables efficient scenario extraction, critical scenario creation, and large-scale virtual validation. By addressing both software infrastructure and scenario generation, Siemens contributes to enhancing the reliability and safety of ADAS and AV systems.

Keywords: autonomous vehicles, virtual verification and validation, scenario-based testing, workflow, toolchain

1. Introduction

The market of Advanced Driver Assistance Systems (ADAS) and Autonomous Vehicles (AVs) is growing, driven by advancement in sensor technologies, artificial intelligence and the need of safer and efficient transportation. This section introduces the market of ADAS and AV, main challenges associated with AV development, scenario-based testing and overview of the proposed approach.

1.1 ADAS and AV market and main challenges

Incremental development and acceptance of automated driving technology currently lead to rapid adaptation of ADAS and AV technology. These systems provide enhanced safety features and a higher level of automation. The ADAS features of lane-keep assist, adaptive cruise control, emergency braking, and traffic sign recognition have contributed to achieving the different levels of vehicle automation. Finally, with the AV technology, the vehicle takes control of driving tasks, reducing the driver to a passenger to some extent.

Many vehicle manufacturers and suppliers have invested resources in developing AV technology as there is a market demand, particularly with shared mobility services, public transport, and logistics. Many research findings have predicted that the market for ADAS and AV will keep growing in the upcoming years.

One of the market research reports shows that the market will grow by 11.9% [1]. Another research by Mckinsey and Company provides the market prediction for the upcoming years, shown in **Figure 1** [2]. This research emphasizes growth in the ADAS and AV market up to 300 billion dollars by 2035 with level 4 automation responsible for the largest market share.

Furthermore, over the past few years, advancements in networks and technology have enhanced communication between the vehicles and infrastructure. Connected, Cooperative, and Automated Mobility (CCAM) has broadened the autonomous driving capabilities by providing transport facilities to people who cannot drive, delivering goods when human mobility is restricted, and in remote areas [3].

CCAM also has the capability to improve urban transportation efficiency by minimizing accidents and reducing CO_2 emissions. In a holistic view, CCAM relies on intelligent infrastructure that includes smart roadside units (RSUs) and Vehicle-to-Everything (V2X) communication. High speed and real-time communication is necessary between the vehicles and their surroundings and also traffic telematics - which collects, processes, and utilizes the traffic data and provides it to the relevant autonomous vehicles.

These features will provide vital information such as traffic jam warnings, accident warnings, better alternate routes, taxi information, parking lot information, etc. On the other hand, the advanced technology of AVs faces different challenges, among which some of the major challenges are:

1. *Scalability and interoperability* – To scale the AV operations across different regions of the world, AV systems must be capable of operating safely across different environmental and traffic conditions. The systems should be efficient

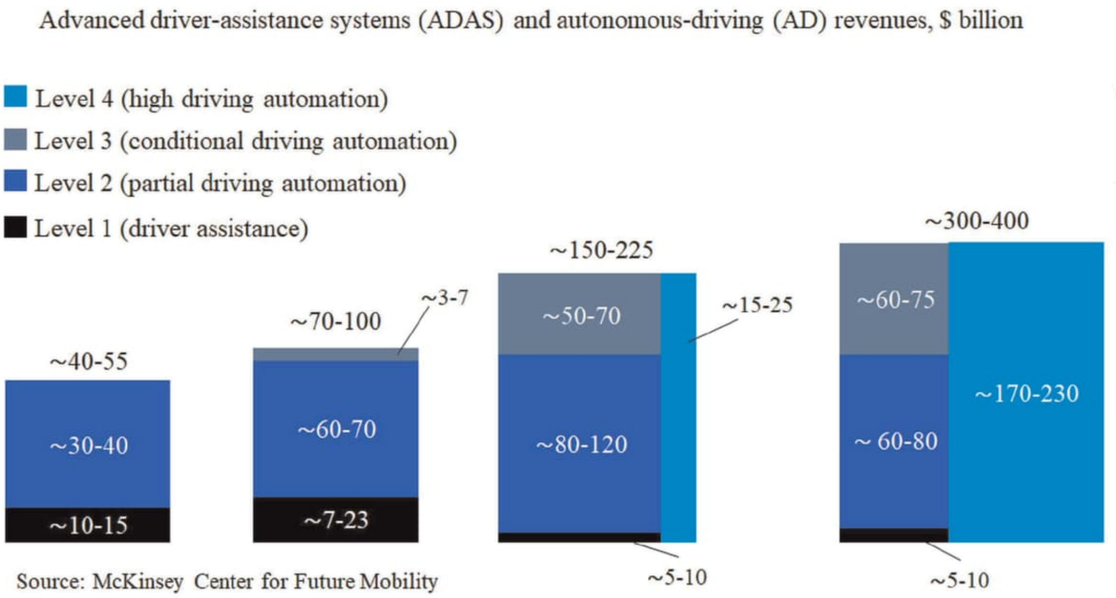


Figure 1.
ADAS market research report.

in delivering the objective of driving themselves considering the unpredictable behavior of the surrounding world.

2. *Efficient testing* – Traditional testing methods involve real-world driving which can be time-consuming, expensive, and might also be limited in the scope of scenario-based testing as the testing team can only perform a fixed number of scenarios with the actual vehicle in the given amount of time. To tackle this, the industry is moving toward simulation and virtual validation techniques. These allow for accelerated testing cycles with opportunities to create a vast number of scenarios. These virtual tests can include edge cases like complex traffic intersections and extreme weather, and all the regular ADAS features, thus making the development envelope larger while decreasing the costs and development time.

3. *Infrastructure* – The current road networks and communication infrastructures are not capable of supporting the advanced requirements of autonomous vehicles. While the sensors on these vehicles are advanced enough to provide comprehensive detection and interpretation of the surroundings, the sensors in the outside world need to be further developed to help in Vehicle-2-Everything (V2X) features. The road markings and traffic signs have to be maintained well throughout the lifetime which are crucial in providing necessary information to the sensors of the AV.

1.2 Introduction to the concept of scenario-based testing and virtual validation

Autonomous vehicles need to navigate safely through different traffic and environmental conditions. It can take a considerable amount of time to test all the possible scenarios with physical testing. To tackle this, *via* simulations, a wide variety of traffic scenarios, as well as different weather and illumination conditions, can be performed in an efficient manner. Engineers and developers can build the scenarios to be tested in the dedicated software, where a digital twin of the actual autonomous vehicle will be deployed and tested.

Scenario-based testing ensures that the AV systems can be rigorously tested across different sets of conditions, improving the performance of perception, planning, and control systems in various aspects. Virtual validation enables rapid iteration and assessments in the virtual environment which will bring down the development time and the related costs. This is achieved by using powerful computing systems to test multiple scenarios continuously where the developers can intervene to improve the controller algorithm or to calibrate the sensors.

Scenario-based testing provides evidence of the reliability and safety of AVs. Virtual validation further enhances this process by enabling continuous testing and iterative development without the additional time and expense of physical prototypes. Together, these help ensure that the AV technology is rigorously tested and refined, before deploying onto physical hardware.

1.3 Overview of proposed approach and main contributions

Siemens provides scalable solutions for safety validation with a portfolio of software tools and an integrated toolchain that complies with international regulations and standards. By using recorded data to extract critical scenarios and synthetically generate edge cases, the toolchain enables automatic and effective scenario synthesis. Taking care of the software infrastructure and adding scenario-generation elements expedites the validation procedure and reduces the safety hazards in ADAS and AV systems.

Siemens' approach for safety validation of automated vehicles is showcased in **Figure 2**. The workflow consists of different steps that commence with requirements management, followed by scenario extraction and creation, and finally the assessment phase. This workflow, together with the toolchain, systematically supports the scenario-based testing of ADAS and AV systems using Model-in-the-Loop (MiL), Software-in-the-Loop (SiL), and Hardware-in-the-Loop (HiL) setups.

The main contributions of this book chapter can be summarized as follows. Section 2 deals with deriving and managing the requirements. First, the applicable legislation as well as the relevant standards are highlighted. Next, the first steps of the overall workflow are described, starting with the description of the operational design domain, definition of the dynamic driving task and requirements management. All these steps, define requirements for data collection and scenario extraction/creation, which are described in next section.

Section 3 introduces a systematic approach to extracting known-safe scenarios from real-world data using Siemens' Simcenter Autonomy Data Analysis (ADA) tool. These scenarios serve as the foundation for further testing and optimization. Siemens' Critical Scenario Creation (CSC) tool innovatively transforms known-safe scenarios into high-risk scenarios, including known-unsafe and unknown-unsafe situations. The use of advanced optimization techniques for scenario creation addresses potential risks and improves testing comprehensiveness.

The toolchain incorporates standardized test scenarios from regulatory frameworks such as EURO NCAP, UN-R131, and ISO standards, ensuring compliance with safety and performance regulations for autonomous and advanced driver-assistance systems.

Section 4 presents the Siemens methodology and tooling for virtual scenario-based testing, including high-fidelity simulation models, advanced techniques for scenario selection and sampling, and methods for requirements coverage. The methodology and tooling are also discussed with respect to simulation credibility assessment frameworks, providing examples of usability and simulation models and results validation.

Section 5 discusses the verification and validation of ADAS and AV using simulation. In particular, Software-in-the-Loop (SiL) and Hardware-in-the-Loop (HiL)

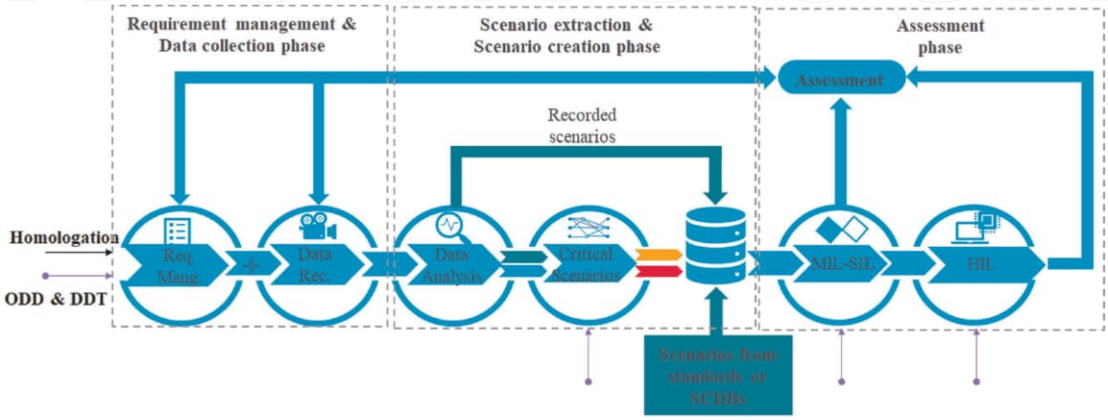


Figure 2.
Siemens autonomy toolchain and workflow for ADAS/AV.

environments and CCAM testing are discussed. The challenges of these setups are summarized and two examples of HiL simulations are given to illustrate these challenges and to propose solutions for them. Finally, Section 6 summarizes the main conclusions of this chapter.

2. Deriving and managing requirements for ADAS/AV

In this section first the applicable legislation and standards are discussed, then the operational design domain description is provided and finally, the importance of requirement management and tracking is discussed.

2.1 Applicable legislation and standards for ADAS/AV

In August 2022, the EU Commission adopted regulation 2022/1426 laying down rules for the application of regulation (EU) 2019/2144 of the European Parliament and of the Council as regards uniform procedures and technical specifications for the type-approval of the automated driving system (ADS) of fully automated vehicles.

The assessment of the automated driving system of fully automated vehicles, as proposed by this regulation, relies heavily on the traffic scenarios that are relevant to the different use cases of fully automated vehicles. It is therefore necessary to define those different use cases.

Given the complexity of automated driving systems, it is necessary to supplement the performance requirements and tests of this regulation with manufacturer documentation demonstrating that the automated driving system is free of unreasonable safety risks to vehicle occupants and other road users in the relevant scenarios and during the ADS lifetime.

In this sense, in February 2021, the United Nations Economic Commission for Europe (UN-ECE) presented the New Assessment/Test Method for Automated Driving (NATM) [4] – a framework, that introduces a multi-pillar approach for safety validation of automated driving, see **Figure 3**.

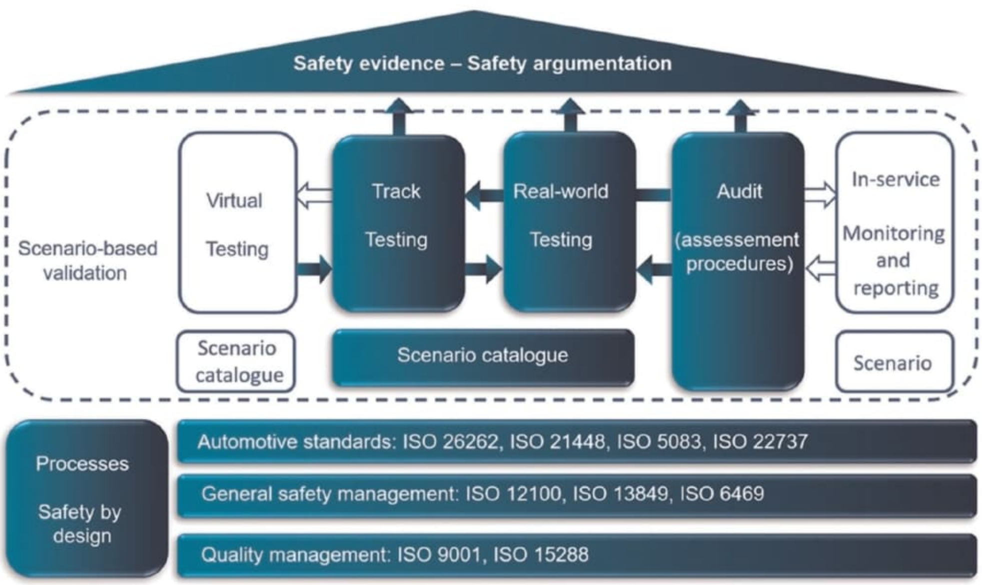


Figure 3.
Multi-pillar approach for safety validation of automated driving systems.

The multi-pillar safety validation of automated vehicles specifies five certification pillars, which support the safety argumentation. In addition to the three well-known pillars (track testing, real-world testing, and audit), the regulation mentions virtual testing and in-service monitoring.

In this document, the verification, validation, assurance, and certification are defined/described as follows:

- Verification is an activity that determines whether a system meets the requirements, answering the question: "Did we build the system right?"
- Validation is assessing if the system meets the end user needs, answering the question: "Did we build the right system?". On the other hand, model validation is evaluating how well the model represents reality.
- Assurance is justified confidence that the system functions as intended.
- Certification determines whether a system conforms to a set of criteria or standards.

In the next subsections, we briefly overview the most relevant standards.

2.1.1 ISO 26262: Functional safety standard

The ISO26262 functional safety standard is well-known and widely used in the automotive sector [5]. The standard lays down the main requirements of how the system should detect and respond to failures, errors, or off-nominal performance. As shown on the left side of **Figure 4** ISO 26262 defined several workflows such as product development at the system level, software development, and hardware development. Each workflow includes testing, verification, and validation activities.

The detailed software development workflow, including software integration and testing, verification of software safety requirements, system integration, and testing is presented in **Figure 4**, see right side.

2.1.2 ISO 21448: SOTIF

The ISO 21448 (SOTIF) – safety of the intended functionality – describes how the system should detect and respond to functional insufficiencies of the intended functionality or reasonably foreseeable misuse by persons [6].

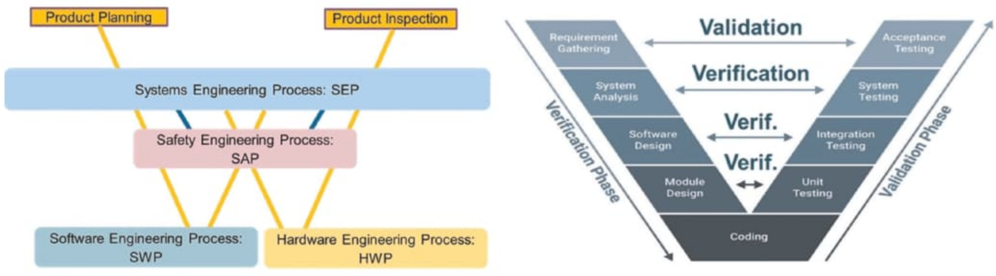


Figure 4.
Development workflow (left) and software development workflow (right) according to ISO 26262.

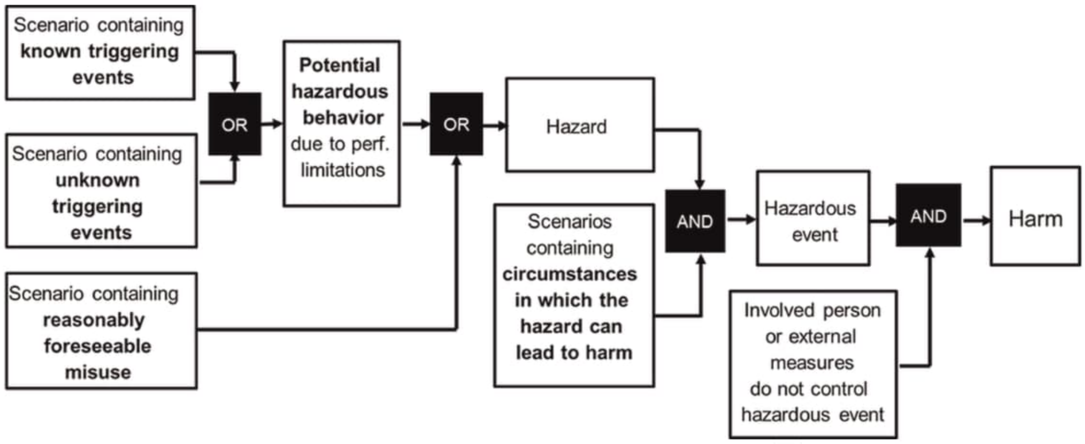


Figure 5.
Hazardous event model according to ISO 21448.

The objective is to validate the automated function in all relevant scenarios, especially under difficult conditions for both sensors and algorithms. As a remark: functional insufficiencies at the vehicle level due to - insufficiency of specifications (e.g., incorrect/incomplete specifications) - performance limitation (e.g., limited sensors range, overestimated braking assistance).

The end goal – SOTIF release – is the absence of unreasonable risk due to hazards resulting from functional insufficiencies of the intended functionality or by reasonably foreseeable misuse by road users. The hazardous event model according to ISO 21448 is presented in **Figure 5**.

ISO 21448 introduces the concept of scenario-based testing, where the scenario is defined as a sequence of scenes usually including the automated driving system(s) (ADS)/subject vehicle(s), and its/their interactions in the process of performing the dynamic driving task (DDT).

Furthermore, the scene is defined as a snapshot of all entities including, but not limited to the automated driving system (ADS)/subject vehicle, scenery, dynamic environment, and all actors and observer's self-representations, and the relationships between those entities. Finally, scenery is defined as part of the environment that remains unchanged.

According to ISO 21448, the development is a progressive process as shown in **Figure 6**. The scenario space is divided into safe and hazardous (non-safe) scenarios as well as known and unknown scenarios.

One of the major goals of the development according to ISO 21448 is to discover the unknown and hazardous scenarios and make them known and later by certain technical measures (e.g., by adapting the system under test of limiting the operational design domain) make them safe. Finally, the "area" of safe and known scenarios shall increase and the area of hazardous scenarios shall decrease significantly.

Other relevant standards, which are not detailed here, include ISO 22737 and ISO 34502. The ISO 22737 standard specifies low-speed automated driving (LSAD) system requirements and procedures, which are going to assist manufacturers of LSAD systems in the incorporation of minimum safety requirements in their designs and allow end users, operators, and regulators to reference a minimum set of performance requirements in their procurements [7].

ISO 22737 is a very clear and prescriptive standard, which can be used during the development of LSAD.

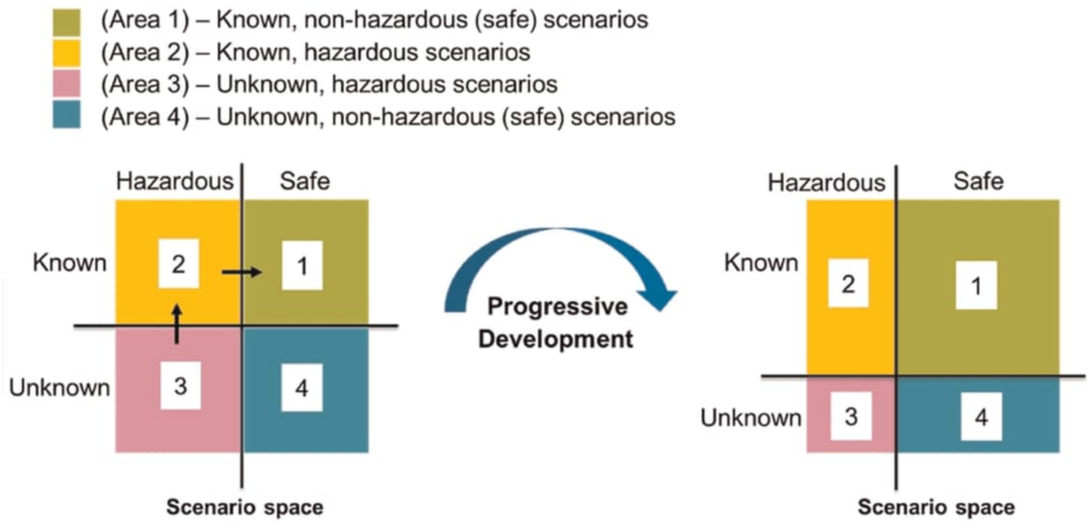


Figure 6.
Progressive development according to ISO 21448.

2.2 ODD description

A key aspect of the safe use of automated vehicle technology is defining its capabilities and limitations and clearly communicating these to the end user, leading to a state of "informed safety". The first step in establishing the capability of an ADS is the definition of its operational design domain (ODD). The ODD represents the operating environment within which an ADS can perform the dynamic driving task (DDT) safely.

As shown in Section 1.3 the verification and validation workflow starts with a description of ODD, a definition of the DDT, and requirements elicitation according to applicable legislation and standards, as shown in **Figure 7**.

In this subsection the focus is mainly on the ODD, the next subsection is related to requirements definition and requirements management, especially how the requirements are linked to the assessment. Based on the ODD definition and defined requirements the data collection/recording can start, which can feed the scenario generation as well as the model validation process.

The ODD taxonomy specified in this BSI PAS 1883 enables ADS manufacturers to specify and implement minimum safety requirements in their designs, and allows end users, operators, and regulators to reference a minimum set of ODD attributes and performance requirements in their procurements [8].

The ODD description is structured according to **Figure 8** left side and a so-called ODD checklist is included: see **Figure 8** right side which is a very important checklist during the design as well as testing of ADS.

It will also enable ADS manufacturers, developers, and suppliers of components and sub-components to define the operating capability and assemble sets of evidence that will improve confidence in the safety of the resulting product (such as component specifications) and in the data obtained from appropriate test and verification activities.

This BSI PAS 1883 document is intended for organizations developing safety cases for automated vehicle trials and testing, manufacturers and developers of Society of Automotive Engineers (SAE) Level 3 and SAE Level 4 ADS, and suppliers of components and sub-components. It is also of interest to insurers, regulators, service providers, and national, local, and regional governments to enable them to understand possible ADS deployments and capabilities.

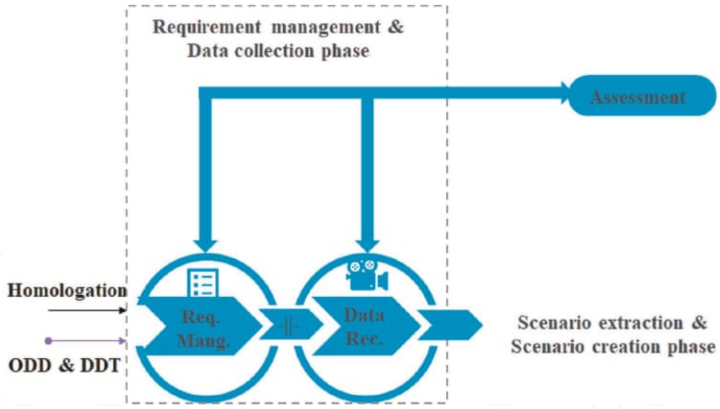
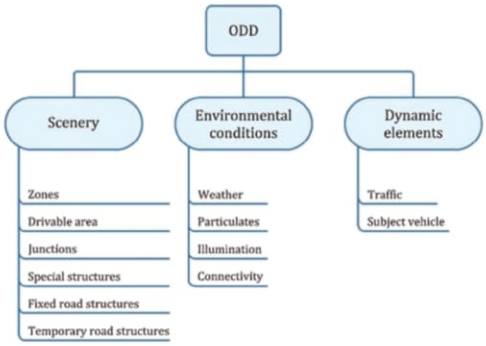


Figure 7.
The first steps of the workflow - requirements management and data collection.



Attribute	Sub-attribute	Sub-attribute	Capability
Drivable area geometry	Horizontal plane	Straight roads	Yes
	—	Curves	Yes – up to 1/500 m (radius of curvature)
	Vertical plane	Up-slope	Yes
		Down-slope	Yes
		Level plane	Yes
	Cross-section	Divided/undivided	Divided
		Pavement	Yes
		Barrier on the edge	No
		Types of lanes together	Only traffic lane

Figure 8.
ODD description (left side) and ODD checklist (right side) according to BSI PAS 1883.

The BSI PAS 1883 document does not cover the basic test procedures for attributes of the ODD, the monitoring requirements of the ODD attributes, and the format of the ODD definition.

2.3 Requirements management

Before we start to discuss requirements management it is essential to define the main characteristics of well-written requirements. Well-written requirements shall have (at least) the following characteristics (see ISO 26262): complete, consistent, feasible, modifiable, unambiguous, and testable.

Complete: The requirements must be complete, meaning they shall contain all the required information to realize/implement the requirement. There is no need to assume anything to realize/implement the requirement.

Consistent: Consistent requirements mean that there is no contradictory information in the requirements document. Feasible: This is one of the crucial aspects of requirements. Requirements shall be implementable within the given time frame and budget and implementable using the existing and chosen technology platform.

Modifiable: In most projects, requirements are never static and do not stop after the requirements document is signed off. The best way to manage the requirements is to manage these changes using a requirements management tool. In case of any changes, the specific requirements and the dependent ones can be modified accordingly without impact on the others.

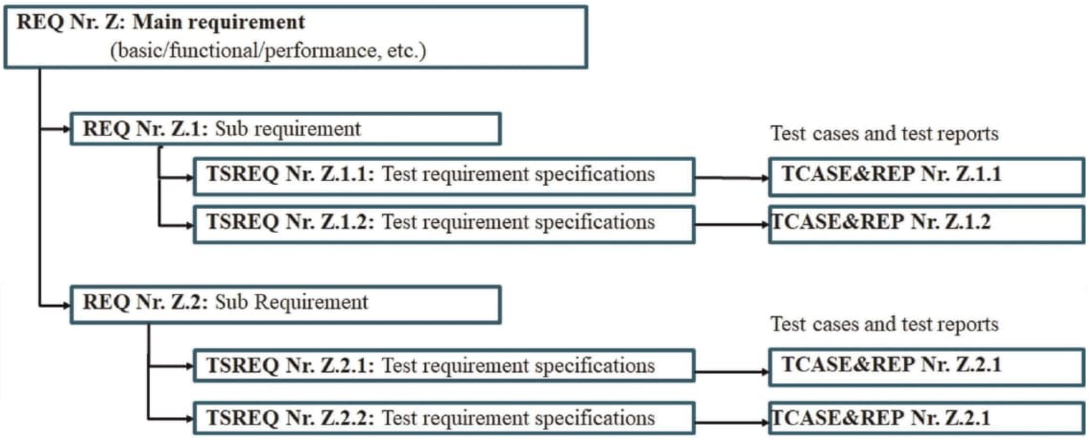


Figure 9.
Requirements, test requirements, test specifications, test cases, and test reports.

Unambiguous: Unambiguous means a single interpretation. If a requirement is defined so that it can only be interpreted in one way, it means it is unambiguous. All subjective words or statements must be eliminated from the requirements.

Testable: A testable requirement can be defined as a requirement, which can be tested and validated using any of the following methods: inspection, walk-through, demonstration, or testing.

Furthermore, the workflow associated with the virtual verification and validation methodology shall be traceable (see continuous integration/continuous deployment ISO 21448), platform-independent, scalable (shall be able to handle large amounts of data) and modular to handle high-complexity systems.

The requirements usually are grouped into main categories, like basic or general requirements, system requirements, functional safety requirements, performance requirements, etc. Furthermore, the requirements are split into sub-requirements, from which test requirements specifications and test cases are derived, see **Figure 9**. These links usually are created manually by the requirement engineer.

The requirements related to the virtual verification and validation (V&V) methodology shall clearly separate verification requirements and validation requirements.

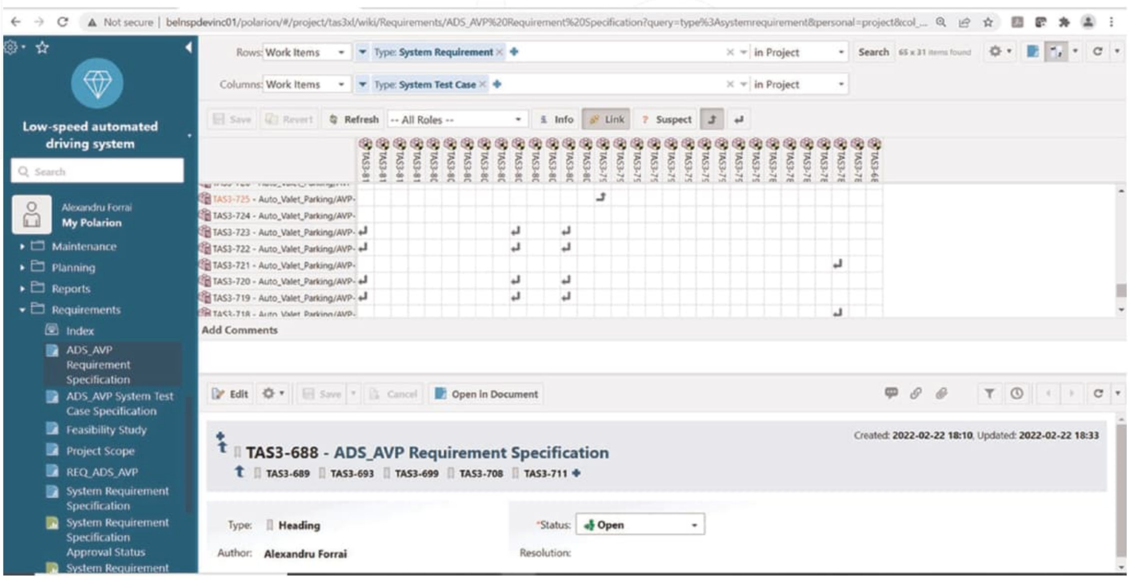


Figure 10.
Traceability matrix – linking requirements with test cases – example in polarion.

Requirements shall be linked to test cases, so traceability between requirements and test cases is established (see ISO 26262).

Subsystems and systems are verified against technical requirements (in line with the concept "doing things right") and subsystems and systems are validated against end user requirements ("doing the right things"), see ISO 26262.

By linking the requirements with the test requirements and test cases, a so-called traceability matrix is generated as shown in **Figure 10**, where a requirement shall be tested by at least one test requirement.

3. Scenario extraction and creation

This section outlines the process of extracting and creating scenarios for testing autonomous vehicle systems. First, known-safe scenarios are extracted from real-world driving data using the Simcenter ADA tool. These scenarios form the foundation for further testing.

To assess system performance under more challenging conditions, Siemens' Critical Scenario Creation (CSC) tool transforms known-safe scenarios into known-unsafe scenarios by optimizing risk factors. The CSC tool also creates unknown-unsafe scenarios through a three-step process involving feature extraction, behavior configuration, and risk identification, using advanced optimization techniques. Additionally, standard test scenarios from regulatory bodies are integrated to ensure compliance with safety protocols and regulations.

3.1 Data collection and data analysis

The data collection process begins by gathering information from various sources, including vehicle sensors, drone footage, and infrastructure-mounted devices, to capture real-world driving scenarios within the ODD. Siemens provides the necessary tools and services for sensor setup, data recording, and data processing.

This process ensures that actor behaviors, traffic flow, and other environmental elements are accurately captured and processed. Errors in the collected data, such as noise, missed detections, and occlusions, are corrected during data processing, which results in reliable and complete trajectory datasets.

In terms of data analysis, the Simcenter ADA tool is responsible for processing raw data and extracting known-safe scenarios. Simcenter ADA analyzes the data to extract, replay, and categorize scenarios, evaluating them based on key performance indicators (KPIs). Additionally, users can explore specific scenarios in detail and export the data for further simulation.

However, Simcenter ADA only extracts known-safe scenarios, which are essential for testing under noncritical conditions. These safe scenarios serve as a baseline for further critical scenario generation.

3.2 Scenario extraction

Scenario extraction is primarily handled by the Simcenter ADA tool, which identifies known-safe scenarios based on safety thresholds. These scenarios are carefully categorized to ensure that they meet the requirements. Known-safe scenarios are valuable for verifying system performance in stable, nonhazardous conditions.

To address critical testing needs, Siemens also employs its Critical Scenario Creation (CSC) tool, which builds upon the known-safe scenarios extracted by Simcenter ADA. The CSC tool is used to generate two types of high-risk scenarios: known-unsafe and unknown-unsafe.

3.3 Critical scenario creation

The Critical Scenario Creation (CSC) tool plays a crucial role in transforming known-safe scenarios into high-risk scenarios that are essential for testing autonomous vehicle systems in more dangerous and unpredictable situations.

- Known-unsafe scenarios: Based on the known-safe scenarios provided by Simcenter ADA, the CSC tool optimizes scenario parameters to identify potential risk factors. By increasing the criticality of the known scenario, the tool generates known-unsafe scenarios (shown by the orange arrow in **Figure 11**). These scenarios are critical for assessing how well autonomous systems handle predictable, hazardous conditions.

- Unknown-unsafe scenarios: The creation of unknown-unsafe scenarios involves a more complex process, as these scenarios represent unforeseen, potentially dangerous situations that may not have been previously encountered. The CSC tool provides a three-step process for unknown-unsafe scenario creation involves ([9, 10]):

1. Extraction: The first step is to extract key features that describe the behavior of actors in the scene, such as vehicle paths, velocity profile and offset to center of the lane. These features are determined by modeling the road layout as a graph and calculating probability distributions for each parameter and node combination. For actor i, the probability of behavior $P_{a,i}$ is computed as:

$$P_{a,i} = P_{p,i} \prod_{j=1}^{m} P_{par,j} \tag{1}$$

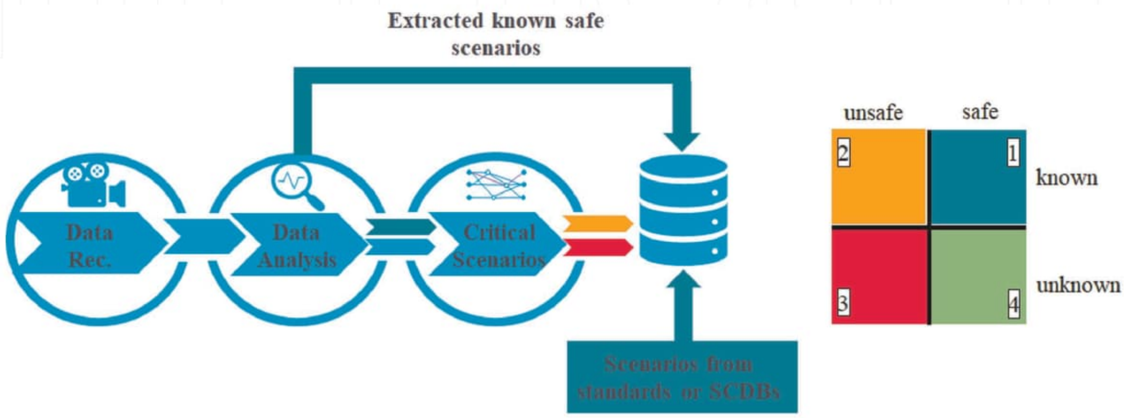


Figure 11.
SOTIF scenarios (right), Siemens autonomy toolchain for scenario generation (left), and arrows in green show scenarios from safe-known, orange, and red arrows show the known-unsafe and unknown-unsafe scenarios, respectively.

where:

- $P_{p,i}$ is the probability of a path cluster for actor i,
- $P_{par,j}$ represents the probability of the $j-th$ parameter,
- m is the number of parameters extracted for actor i.

1. Configuration: The next step is to configure the behavior of actors and parameters to define the search space. Given the complexity of the scenario, the tool automatically reduces the search space by identifying noninteracting actor paths with ego vehicle and partitioning the remaining space based on discrete parameters, for example, colliding actor type. This step reduces the search space, making it manageable for further optimization.
2. Identification: The final step involves applying an optimization algorithm to assess the risk and novelty of the generated scenarios. This step uses proprietary metrics to identify the most critical scenarios, focusing on time to collision (TTC) and unexpectedness. The unexpectancy is calculated based on the difference between what the ego vehicle predicts and what is happening in a scenario. The objective function $f(\eta)$, which evaluates the risk of a scenario (having scenario parameters η) is defined as:

$$f(\eta) = G(P_s(\eta))(2 - \epsilon(\eta) + TTC_{min}(\eta)) \tag{2}$$

where:

- $G(P_s(\eta))$ is a function of the scenario probability $P_s(\eta)$,
- $\epsilon(\eta)$ is an unexpectedness metric to measure novelty (with a constant of two added to ensure a nonnegative value),
- $TTC_{min}(\eta)$ represents the minimum time to collision.

The scenario probability $P_s(\eta)$ is calculated as:

$$P_s(\eta) = \lambda_{TL} \prod_{i=1}^{n} P_{a,i} \tag{3}$$

where:

- λ_{TL} is a factor in traffic lights,
- $P_{a,i}$ is the probability of actor behavior i,
- n is the number of actors.

The optimization problem is formulated to minimize the risk function $f(\eta)$, subject to constraints:

$$\min_{\eta} f(\eta) \qquad subject\ \ to \qquad \eta \in X, \zeta_{col} = 0 \tag{4}$$

where X defines the bounds for the scenario parameters, and ζ_{col} is a Boolean variable that is 0 in case of ego collision and 1 in case of non-ego actors collision. This approach allows the CSC tool to generate unknown-unsafe scenarios (as shown by the red arrow in **Figure 11**), which represent unforeseen, potentially dangerous situations. These scenarios provide critical insights into how autonomous systems respond to rare and unexpected hazards.

3.4 Standard scenarios

In addition to the scenarios generated through extraction and critical scenario creation, the Siemens toolchain also incorporates standardized test scenarios from various regulatory bodies and accident databases. These scenarios are based on industry-recognized standards, including:

- EURO NCAP (for Advanced Driver Assistance Systems, ADAS)
- UN R131 and UN R152 (Automated Emergency Braking Systems, AEBS)
- UN R157 (Automated Lane Keeping Systems, ALKS)

For autonomous vehicles, Siemens utilizes test scenarios specified by ISO standards, such as ISO 22737 (LSAD) and ISO/DIS 23374-1 (automated valet parking system), which provide guidelines for verifying and validating the performance of these systems in different driving environments. These standardized scenarios serve as a baseline for testing compliance with safety and performance regulations, ensuring that both ADAS and autonomous systems meet global safety requirements.

In addition to adhering to these regulations, the Siemens toolchain allows for the customization of scenarios based on expert input, enabling developers to test additional requirements specific to their operational design domain. These virtual tests help ensure the safety and robustness of the systems before they are subjected to real-world testing on roads.

4. Scenario-based testing: The workflow

In this section, the workflow for scenario-based simulation testing is described. **Figure 12** presents an example of a scenario-based testing workflow using the Siemens toolchain. The workflow begins with scenario selection from scenario databases (SCDBs) based on system under test (SUT) specifications, such as ODD and test requirements. This is followed by various techniques for scenario concretization and test automation. The ADAS/AV function is then evaluated against the concrete scenarios, and results are generated for the SUT assessment.

The Siemens toolchain provides the following key features to enable this workflow:

- Simcenter High-Performance Engineering Exploration and Design Optimization Software (HEEDS) [11] which provides test automation capabilities along with smart sampling and optimization methods.
- Simcenter Prescan [12] fully supports industry standards, ASAM OpenDrive [13] and OpenScenario-XML [14], for scenario description.

- Detailed and physics-based world model for representative sensor simulation. For example, measured materials are provided for accurate radar sensor simulation (e.g., to capture reflectivity and multi-path phenomena).

- Physics-based sensor simulation of ADAS/AV sensors and high-fidelity vehicle dynamics models.

- User-friendly APIs for incorporating ADAS/AV functions and external vehicle dynamics models for closed-loop scenario execution.

Building, configuring, and validating high-fidelity models for simulation is essential to ensure the reliability of simulation results. This is discussed in Section 4.1. The broader framework for the credibility of the simulation results, which includes model validation, is presented in Section 4.2. The final two subsections discuss scenario selection, sampling, and meeting coverage requirements for scenario-based testing.

4.1 Model building, model validation, and digital twins

Depending on the test scenarios and the underlying requirements, the fidelity required for the simulation models may be determined. Simcenter Prescan provides sensor models and vehicle dynamics models of different fidelity. For example, when testing perception functions, high-fidelity physics-based sensors may be used. On the other hand, when testing planning and control functions, ground truth sensor models or probabilistic sensor models which can output perception results with a certain error rate can be used.

The process of model building and model validation can be illustrated through the example of the physics-based camera (PBC) sensor simulation in Simcenter Prescan. The process of modeling can be separated into three parts: (a) the generic PBC model and camera sensor simulation, (b) configuration of the PBC model to a real camera, and (c) validation of the PBC model against the real camera.

The PBC model consists of a pipeline that models different layers of the physical sensor such as the lens and color filter array. The high-fidelity model can represent various physical effects that may occur, for example, geometric distortion, blooming, and flare. In conjunction with the camera model, the simulation engine must also accurately compute the propagation of light, including reflections, in the scene in different environmental conditions. Together, the physics-based simulation engine and PBC model enable accurate camera sensor simulation.

The high-fidelity PBC model provides many parameters, relating to different physical layers of the camera sensor, to configure the model for real camera sensors. Some relevant information for these parameters can be found in the technical

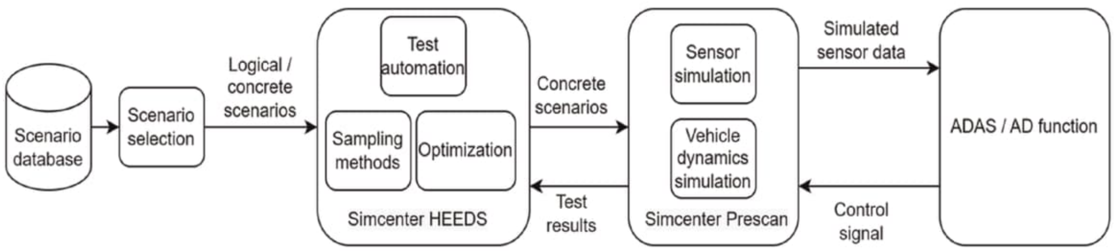


Figure 12.
Example scenario-based testing workflow.

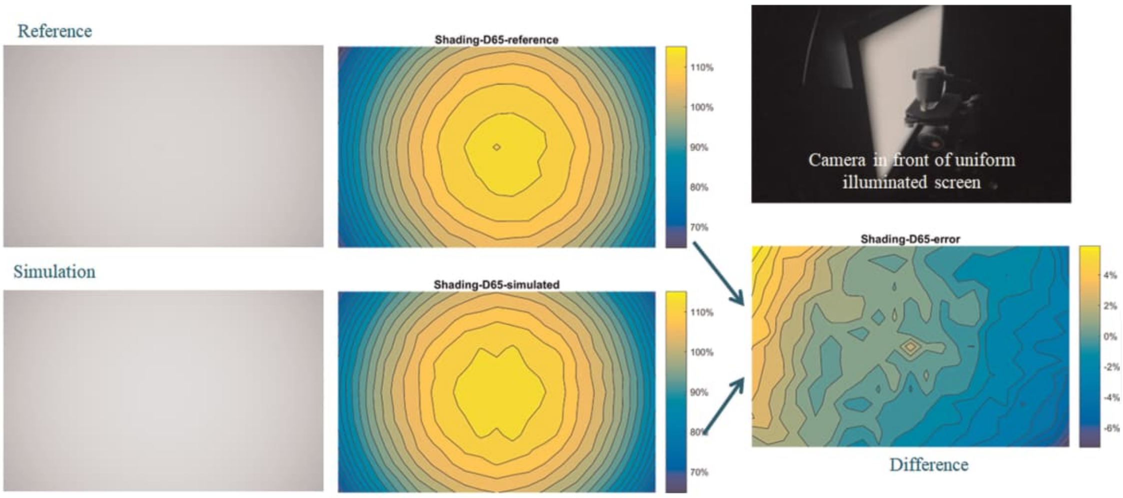


Figure 13.
Example validation of the PBC output against a real camera sensor. Here, the shading property of the camera image is being investigated.

specifications of the sensors. For some important properties, measurements in a laboratory are needed with the physical sensor. For example, a test setup is shown in the top right of **Figure 13**, where the camera is placed in front of a uniformly illuminated screen to understand the shading (or vignetting) properties of the camera.

The figure also shows how lab measurements are used for model validation. For this example, the vignetting behavior of the PBC in Simcenter Prescan is computed by recreating the laboratory test within the simulation. The vignetting behavior is then compared to the measured behavior of the physical sensor. Lab measurements may also be performed to validate sensor simulation under diverse weather conditions.

Additionally, the validation may be performed by evaluating the behavior of the target ADAS/AV camera processing pipeline on the camera simulation model outputs and the physical camera outputs. The test conditions chosen for validation should ideally cover the ODD sufficiently. It is important especially to focus on key factors known to impact the physical sensor and related AD functions, for example, weather and illumination.

4.2 Simulation credibility assessment

As simulation is an essential pillar of ADS safety assessment, it is important to assess and ensure the credibility of simulation results. Technical guidance on the credibility assessment of simulation-based testing is provided in the NATM [4] and the EU 2022/1426 regulation [15].

These documents provide a framework on credibility, which covers topics such as the suitability of the simulation models with respect to the ODD and assessment goals, and the correlation between simulation and physical testing. The credibility framework also includes the model building and model validation processes. These were already described in the previous section (Section 4.1). Below are two other aspects of how the Siemens toolchain conforms to the technical guidance in the above standards.

The usability pillar of the credibility framework includes documenting assumptions and limitations of simulation tooling and preventing incorrect use of the tools. The Simcenter Prescan manual and user graphical user interface (GUI) both provide such measures.

Consider the example of camera sensor simulation in rainy conditions. Firstly, the Simcenter Prescan user manual [12] details the theoretical foundations of the rain model, for example, principles governing the relationship between raindrop fall speed and raindrop diameter. The user is then guided to realistic values for the rain parameters, with recommendations are available also in the GUI. The user is further warned about limitations in accurately rendering the effect of light sources on the intensity of raindrops. To prevent incorrect use, a warning appears in the GUI, as described in the user manual:

> "In this release, weather (rain) is not directly influenced by car/street lights ... (this) will result in a parse warning"

To further demonstrate the simulation model validity, Siemens V&V methodology includes the use of hardware-in-loop testing and track testing to validate the simulation results. A subset of scenarios that have been tested in the simulation are used for a correlation analysis between simulation results and physical testing. The selection of the test scenarios sufficiently covers the ODD, but emphasizes key scenario regions:

- scenarios with results close to safety thresholds,
- known high variance regions of the simulator *(based on previous correlation studies, model validation, or known limitations of the simulator)*.
- safety-critical regions that are identified based on safety standards, for example, triggering conditions found during SOTIF safety analysis of the SUT.

4.3 Scenario selection and parametrization

The Siemens methodology and tooling include automatic retrieval of scenarios from scenario databases (SCDBs) based on use case specifications such as ODD and test requirements. This reduces manual effort and human error when searching SCDBs for scenarios.

For the universality of the tool, input and output formats must adhere to ISO 34503 (ODD taxonomy and specification) [16] and ASAM OpenLabel standards (scenario, behavior description) [17]. The input ODD definition must be specified as per ISO 34503 standard, while the input test requirements and the output query criteria are provided based on taxonomies of ISO 34503 and OpenLabel.

The tool is presented in **Figure 14**. Firstly, the natural language ODD definition (as per ISO 34503) is interpreted into a machine-readable format, and the requirements are parsed using the ISO 34503 ontology reference. Then, scenario query definition is performed per requirement, such that a traceable set of scenarios is obtained per requirement. Here, the test elements necessary based on the ODD and a requirement are identified.

For example, consider a requirement: *crossing traffic at intersections shall be detected by radar in dense fog* and an ODD definition including *fog: [clear, moderate, medium, dense]*. For this example requirement, dense fog is a relevant test attribute, and other ODD fog values are ignored. Once test elements are identified, a search query is generated and passed to the SCDB application program interface (API). The retrieved scenarios are provided to the user, and a traceability matrix is generated. The user is provided feedback when requirements are unclear or under-specified, such that the automated retrieval of scenarios is not possible.

The selected scenarios are then prepared for testing. Simcenter HEEDS provides test automation capabilities and advanced sampling techniques to efficiently test the SUT against the retrieved scenarios. For example, adaptive sampling iteratively samples the design space to increase test efforts in specific regions of interest within the parameter space, such as regions where the SUT behavior may change with smaller parameter steps.

Figure 15 shows how adaptive sampling efficiently approximates the response surface of a SUT in a cut-in scenario with two parameters: the SUT speed (horizontal axis) and cut-in vehicle speed (vertical axis).

The reference response surface (shown on the left) is obtained with an extensive grid-based sampling method. This response surface shows, as expected, that the SUT is unsafe when the ego vehicle has high speed and the cut-in vehicle has low speeds (top-left region of parameter space). However, unexpectedly, the SUT also behaves unsafely for high speeds of the cut-in vehicle (right-hand side of parameter space).

With a limited number of samples, adaptive sampling efficiently approximates the response surface; distinguishing unsafe regions from safe regions and detecting edge cases in the parameter space.

4.4 Assessment of scenario-based testing - workflow and requirements-based coverage

Safety argumentation for ADAS/AV requires test coverage for requirements. In this subsection, an example of effectively combining field testing and simulation-based testing is presented to meet coverage needs. **Figure 16** shows the process by which Simcenter ADA [18] extracts scenarios encountered by the SUT during field testing. A coverage analysis is then performed given the scenarios against the requirement verification plan. Gaps in requirement coverage (e.g., testing under diverse weather conditions) are then met through additional scenario generation or scenario retrieval from SCDBs as explained in the previous subsection. The additional scenarios are tested in simulation.

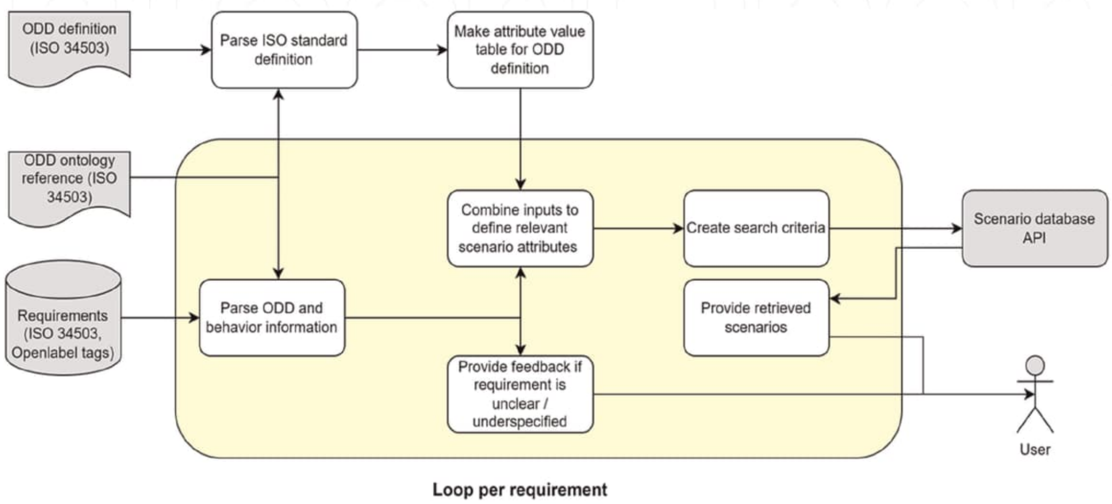


Figure 14.
Automated scenario retrieval from scenario databases.

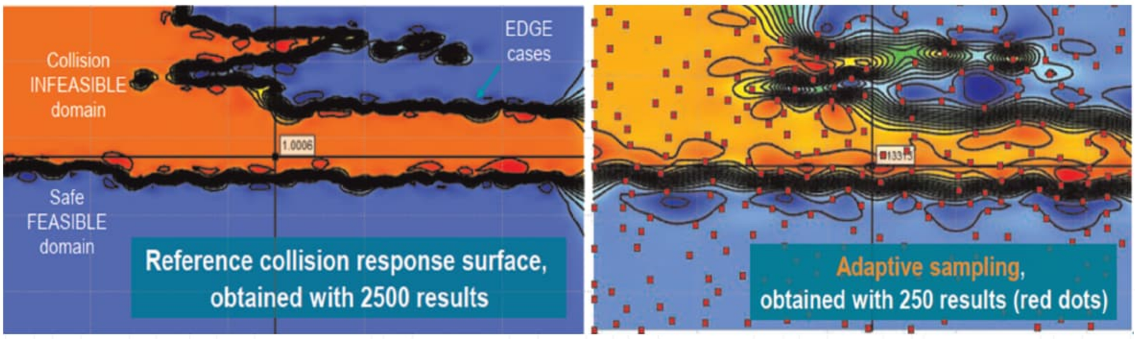


Figure 15.
The left figure shows a reference response surface of an SUT for a cut-in scenario, where the y-axis is SUT speed and the x-axis is cut-in vehicle speed. The blue region is safe, the red region is unsafe. The right figure presents the approximated response surface using Simcenter HEEDS adaptive sampling. The colors represent prediction confidence in safety, ranging from dark blue (safe with high confidence) to dark red (unsafe with high confidence).

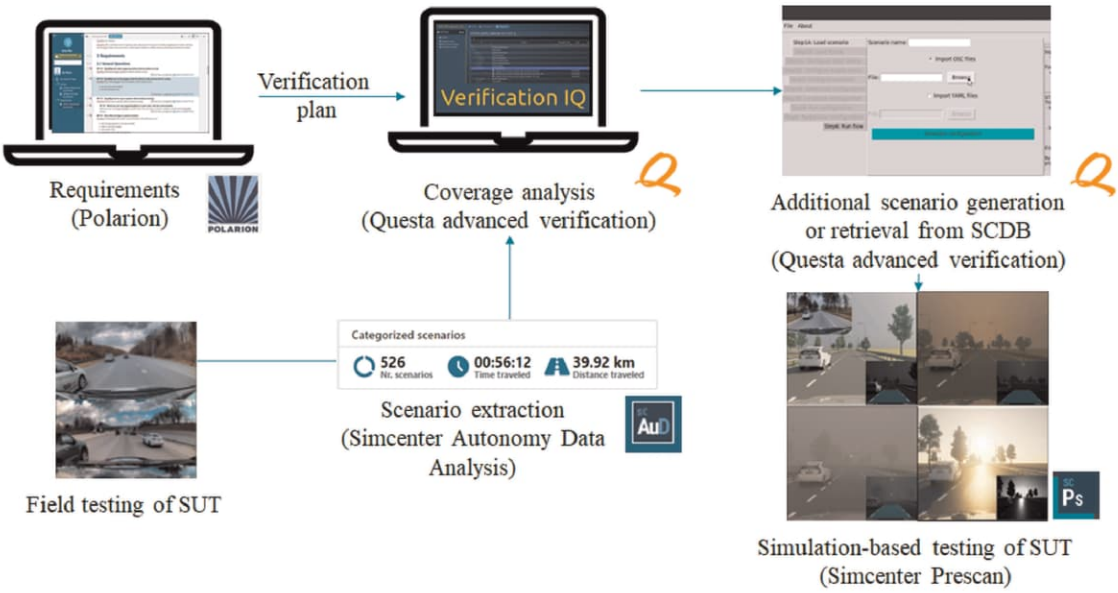


Figure 16.
Workflow for achieving requirements coverage by augmenting field testing with simulation-based testing. The Siemens tools enabling the workflow are mentioned in brackets.

For a full-scale and iterative V&V of ADAS/AV, a good test infrastructure with automated testing based on SUT updates, requirement management, and visualization dashboards to highlight coverage and assessment results is essential. The iterative V&V process often leads to new requirements and further developments of the SUT. The Siemens methodology and tooling presented in this section enable efficiently performing these V&V iterations and creating a solid safety argument for the SUT.

5. Virtual verification and validation

As became clear from the previous section, road vehicle automation functions, both for ADAS and AV, require extensive testing for verification of the requirements and validation of the resulting functionality, among others with respect to SOTIF. The assessment phase of Siemens autonomy toolchain is shown in **Figure 17**.

To enable this assessment efficiently, simulation environments play an important role. Various types of simulation environments exist, as explained in subsection 5.1. One of these is hardware-in-the-Loop (HiL), which is essential during assessment but generally poses one of the biggest technical challenges. This is elaborated upon in subsection 5.2, whereas subsection 5.3 provides two examples of HiL simulations.

Due to the inherent connectivity component in CCAM systems, testing these systems in a HiL-like environment requires specific measures, as further explained in subsection 5.4.

5.1 SiL and HiL testing

To reduce testing costs and time, a shift can be observed from real-world testing to virtual testing. The latter employs high-fidelity simulation models for the vehicle and its environment, while including the ADAS/AV control system in a phased approach, gradually moving from Model-in-the-Loop (MiL) *via* Software-in-the-Loop (SiL) toward the actual hardware implementation in a Hardware-in-the-Loop (HiL) setup following Siemens' toolchain as explained previously and also described in Ref. [19].

Whereas MiL simulations are primarily useful at the development phase of the automation function, SiL and HiL simulations target verification and regression testing of the automation function. Here, SiL focuses on efficiently checking compiled automation software deployed on the simulation platform, after which the next step is to check the actual hardware implementation of the compiled software in a HiL setup.

In the broader context of ADAS/AV development, these simulation architectures fit into the V-model development lifecycle, where HiL simulation bridges the gap between SiL testing and vehicle-level testing. It serves as a crucial intermediate step, allowing for the validation of both hardware and software components in a controlled, reproducible environment before moving to more expensive and complex vehicle prototypes.

To better understand the simulation setup involved in SiL and HiL, the basic logical architecture of a controlled vehicle (whether equipped with ADAS or with AV functions) in simulation is depicted in **Figure 18**. This figure shows the ego-vehicle model (the platform including actuators), which drives in a virtual environment, the latter including the static environment (road network, buildings, and other infrastructure), and the dynamic environment, that is, the other actors in the simulation. This virtual

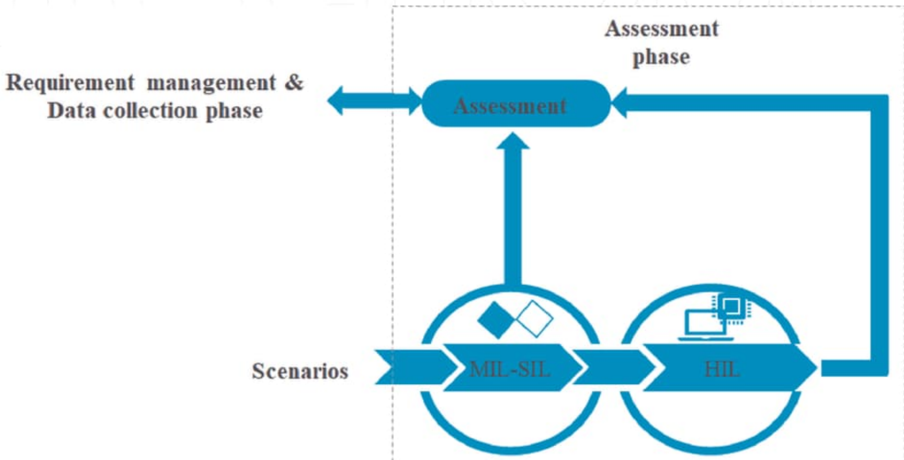


Figure 17.
Siemens autonomy toolchain - assessment phase.

environment is perceived by the ego-vehicle sensors such as lidar, radar, and camera. The output of these sensors as well as the output of vehicle dynamics sensors and, in the case of AV, a mission, constitute the inputs for the ADAS/AV automation software stack involving perception, planning, and control of the ego vehicle.

SiL simulation typically involves a compiled version of the automation software stack, running on a PC, whereas the other simulation components run in a simulation environment such as Simcenter Prescan. SiL simulation is particularly useful for detecting the following types of errors:

- Syntax errors, which already show up at compilation time;
- Runtime errors such as a memory leak, division by zero, and encoding errors;
- Semantic errors, that is, the automation software stack does not generate error messages, but the result of the computations is not what the programmer intended.

As such, SiL simulation serves to discover compilation problems, but not yet implementation problems. The latter is the focus of HiL simulation.

HiL simulation is a technique where real hardware components are combined with a virtual, simulated environment, thus establishing a hybrid testing environment. As ADAS/AV function software reaches the stage of deployment on target hardware, HiL provides a valuable environment for seamless integration and thorough testing of the target hardware in a simulated environment. As a result, HiL setups have become an indispensable tool in the development and testing of ADAS and AV functions.

One of the primary advantages of HiL setups is their ability to facilitate early testing, that is, when access to complete vehicle systems is yet limited or unavailable. By creating a virtual environment that simulates real-world conditions, HiL allows developers and engineers to conduct comprehensive tests, identify potential issues, and optimize performance well before the software is implemented in an actual vehicle, thereby saving time, effort, and costs associated with real-life testing on test tracks and public roads.

Additionally, risks associated with testing ADAS/AV software in real vehicles are minimized, ensuring a higher level of safety throughout the development process. Last but not least, HiL testing also allows for the exploration of edge cases and rare scenarios that might be difficult or dangerous to replicate in real-world testing. It is specifically this characteristic that makes HiL also useful for the verification and validation process after the ADAS/AV function design.

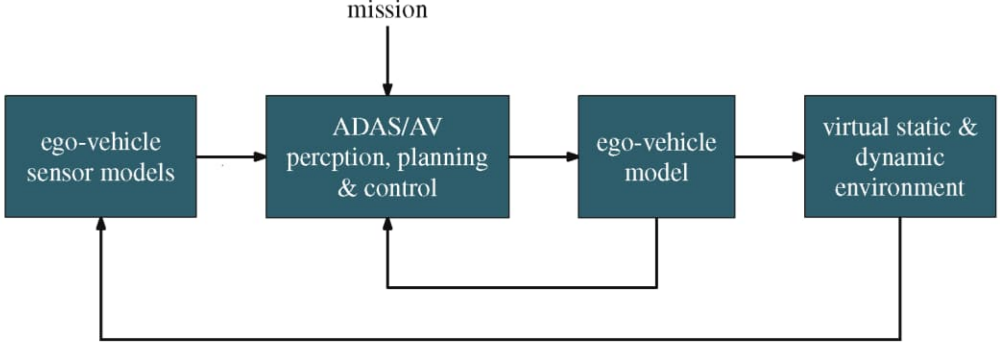


Figure 18.
High-level logical architecture of a vehicle equipped with ADAS/AV functionality in a simulation environment.

5.2 Challenges in HiL simulation

HiL simulations, however, involve several challenges, the most important of which are summarized as follows.

- *Scalability* — State-of-the-art vehicle automation systems employ complex and extensive sensor suites. For instance, Mobileye's SuperVision system [20] utilizes 11 cameras along with a long- and short-range radar sensor. Consequently, HiL setups for these systems must be capable of scaling easily to accommodate a multitude of simulated sensors while maintaining real-time performance.

- *Interfacing* — The hardware under test can be equipped with a large variety of interfaces, often sensor-specific. Examples are GMSL2 for cameras, CAN for radar, and automotive ethernet for lidar. Consequently, the HiL setup has to accommodate those interfaces, which can be considered as a specific aspect of scalability.

- *Synchronization* — Another challenge associated with multi-sensor systems, is the synchronization of data from multiple sensors. Each sensor may have different sampling rates and latencies, requiring algorithms to align the data streams accurately or at least ensure correct time stamping of all data. This synchronization is vital for testing sensor fusion algorithms and ensuring the ADAS/AV system receives a coherent representation of its environment.

To address the above challenges, in particular with respect to scalability, Simcenter Prescan has been designed to meet the demanding requirements of modern HiL setups. One of its key strengths is the inherent capability to be distributed across multiple nodes, providing a robust solution for scaling simulations to accommodate the increasing number of sensors in state-of-the-art ADAS/AV systems. This distributed architecture ensures that the simulation can maintain real-time performance.

Furthermore, Simcenter Prescan offers a powerful feature that allows users to insert custom code for manipulating sensor outputs directly on the GPU or CPU, referred to as User Algorithm on Federate (UAoF). This enables users to modify data streams or redirect them to generic PCIe devices, effectively simulating the physical signals of actual sensors. The ability to do this directly on the GPU or CPU reduces latency in the generation of the physical signals to meet real-time requirements. In summary, the foundations for scalability are created by

- The ability to deploy on multiple machines;

- A mechanism to stream sensor data from the simulation;

- The application of generic PCIe hardware interfaces.

5.3 HiL examples

This section presents two examples of HiL setups. Firstly, a so-called Injection HiL is presented, where simulated sensor signals are injected into an automation software stack running on an Electronic Control Unit (ECU). Secondly, a Projection HiL is described, involving a camera as hardware, detecting a simulated environment projected on a monitor.

5.3.1 Injection HiL

To illustrate the above considerations and the resulting HiL setup, **Figure 19** shows the hardware and software architecture of a setup to test the Apollo AV stack [21] implemented on a high-performance PC that mimics an onboard ECU.

Apollo performs localization, perception, routing, prediction and planning, and control. The inputs consist of vehicle dynamics sensors (termed 'chassis' in the figure), GNSS for vehicle location, and environmental sensor data (lidar and two forward-looking cameras in the figure). The outputs are throttle and brake level, and desired steering angle. Apollo utilizes a middleware layer called CyberRT, which is an open-source runtime framework that is highly optimized for performance, latency, and data throughput.

Next to the Apollo PC, the hardware also involves two PCs to simulate the ego vehicle and its environment using Simcenter Prescan. The first PC, termed Federate 1, simulates the ego vehicle in Simcenter Prescan (in combination with C++) by means of a multi-body Simcenter Amesim vehicle dynamics model, the static and dynamic environment, as well as a lidar model, being one of the onboard sensors.

The second PC, termed Federate 2, is dedicated to running both physics-based camera models since these are computationally intensive. To do so, Simcenter Prescan provides a mechanism to include the static and dynamic environment in Federate 2 during compilation time. Hence, at simulation time, only the actor motion needs to be communicated from Federate 1 to Federate 2 over a TCP connection, thereby reducing data transfer over the network.

Moreover, each camera model includes a UAoF to convert simulation data to the required format (which is RGB to YUV422 conversion in this case). Clearly, this

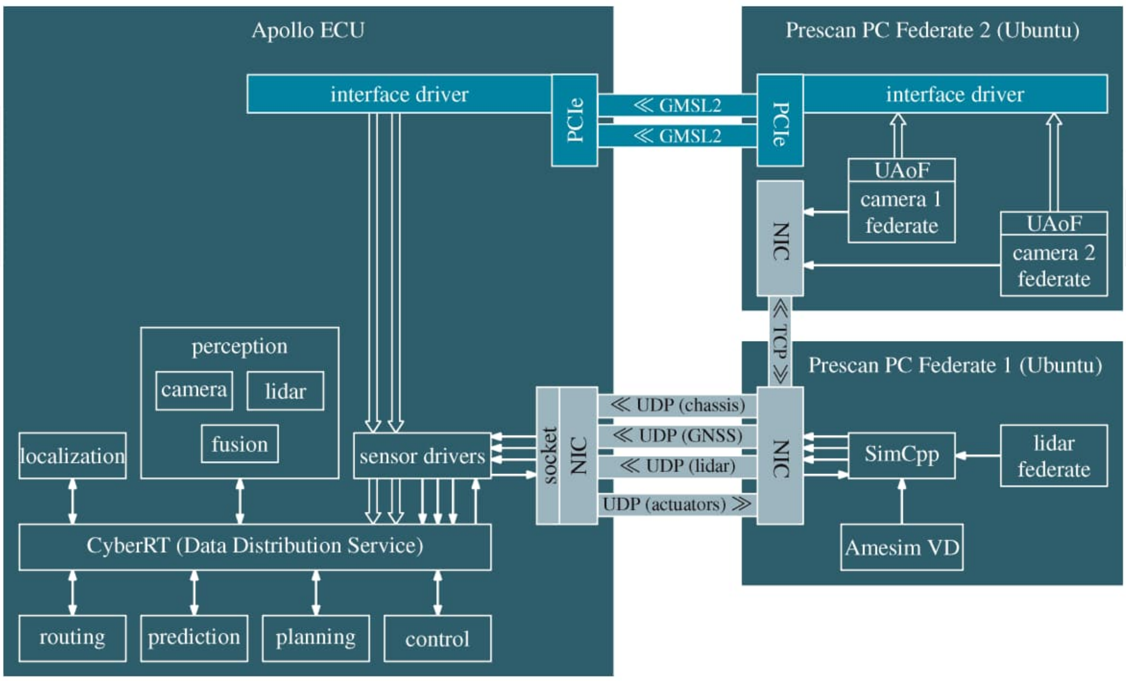


Figure 19.
HiL setup for evaluation of the Apollo AV control system in a virtual environment.

federated architecture introduces scalability with respect to the number of onboard sensors. In fact, any number of federated PC's can be added to run one or more sensor models, thus maintaining real-time execution when scaling up.

The schematic in **Figure 19** also shows the interfaces between the Simcenter Prescan simulation and the Apollo ECU. These interfaces conform to automotive standards, employing GSML2 for camera data and, in this example, UDP over automotive ethernet for lidar and vehicle dynamics sensors, indicated by the connected Network Interface Controllers (NICs) in the figure.

The GSML2 interface is implemented by dedicated PCIe boards. Due to the federated architecture and the UAoF capability, simulation data can be converted to any format that is required by the ECU hardware under test, thereby providing a highly generic setup, capable of accommodating a large variety of ECUs equipped with automotive-grade interfaces.

Finally, it is noted that all clocks in the HiL setup are synchronized with the Precision Time Protocol [22] (not shown in the figure). This allows the sensor fusion algorithm implemented in the ECU to correctly process all incoming sensor signals. Moreover, the Federate PCs execute the time update in a synchronized manner such that the ECU receives a coherent representation of its environment.

5.3.2 Projection HiL

Although HiL is usually associated with testing a hardware implementation of a decision and control system, it is also possible and useful to include other components of the system as depicted in **Figure 18** as hardware in a virtual environment. This is illustrated by the second HiL example, involving an automotive camera as the hardware under test.

The architecture of this HiL setup is shown in **Figure 20**, involving a simulated ego-vehicle, modeled in C++/MATLAB/Simulink, in a virtual environment provided by Simcenter Prescan. One of the ego-vehicle sensors, a camera, is included as hardware-in-the-loop. This camera is facing a monitor which displays the simulated static and dynamic environment. The CAN bus involves bi-directional

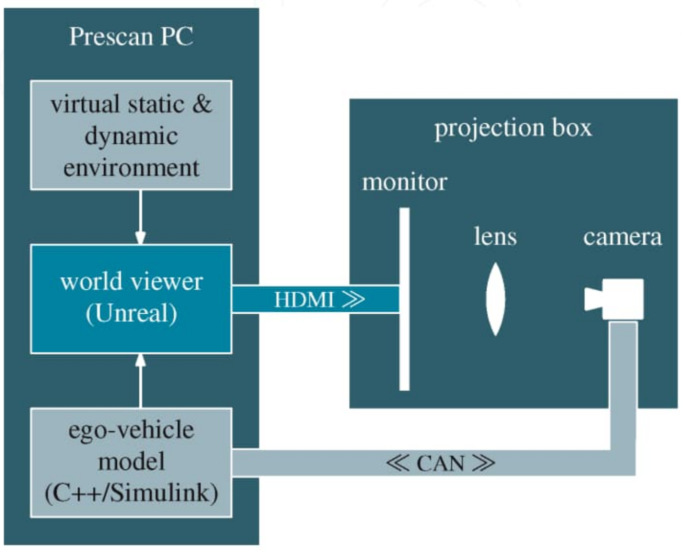


Figure 20.
HiL setup for evaluation of an automotive camera in a virtual environment.

communication, that is, communication of the detected object information to the Simcenter Prescan PC and communication of the ego-vehicle velocity to the camera.

Clearly, this HiL setup is less complex than the previous one. The issue of real-time simulation, however, is nevertheless present: In recent years, automotive cameras show significant improvement in terms of resolution and update rate, moving toward 4 K and 60 Hz, respectively.

Consequently, to provide a realistic simulated environment, the simulation must generate the image frames at the same or higher specifications. At the same time, environment visualization needs to resemble reality to an increasing extent over the past years, in particular, due to requirements imposed by AI-based image processing solutions.

As a result of these developments, the image generation involves an increasing computational effort. To significantly relieve this computational burden, Simcenter Prescan includes the feature of Deep Learning Super Sampling (DLSS), developed by NVIDIA [23]. This feature allows for generating camera frames at a lower resolution, thereby decreasing the rendering effort, after which the required resolution is obtained by low-effort up-sampling using a neural network.

The newer versions of DLSS add AI-based frame generation to the super sampling feature. This involves generating a new frame between two existing frames, hence doubling the update rate. As a result, the images can still be generated according to ever-increasing specifications.

5.4 CCAM testing

Remote monitoring is an essential feature of AVs and CCAM, for monitoring the ODD, performance, and safety monitoring of the automated driving system as well as for monitoring the traffic and the infrastructure. The autonomous vehicles can benefit from being remotely monitored so that if needed, the engineers and developers can look into the recordings and improve the driving performance of the vehicles.

Figure 21 shows the modular approach for remote monitoring. The two systems communicate with each other over the Message Queue Telemetry Transport (MQTT) cloud network.

The vehicle sends image frames, GPS coordinates, and other sensor information to the MQTT cloud. The local server, digital twin (DT) service runs programs to receive the data from the cloud, decode the data, and provide relevant visualization along

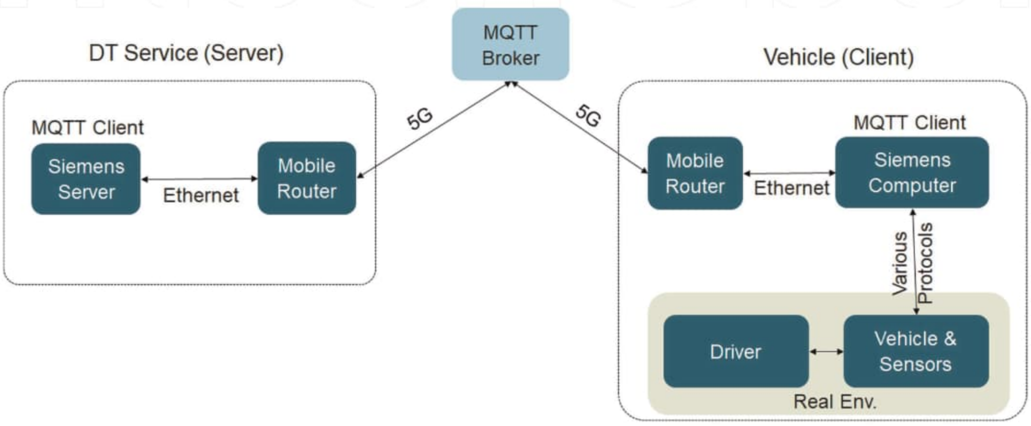


Figure 21.
Remote monitoring essential feature of CCAM and AV.

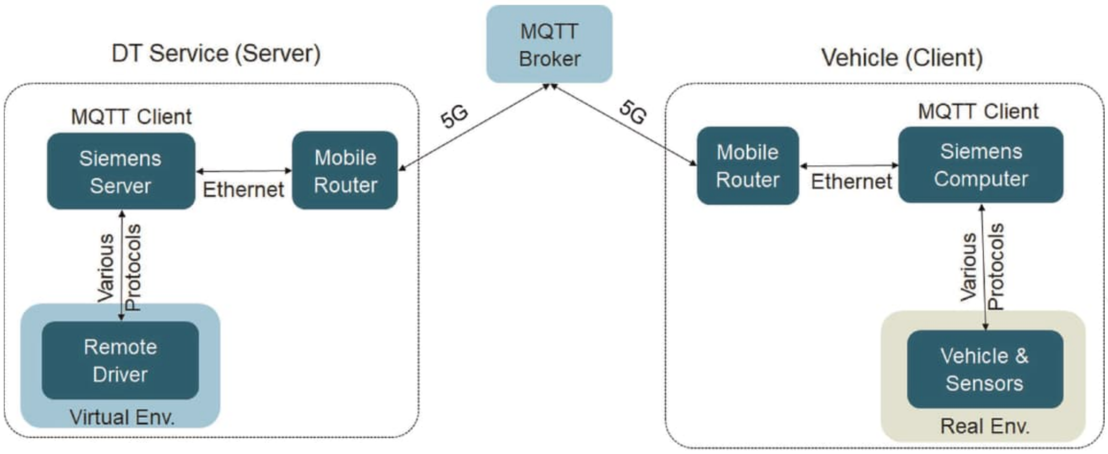


Figure 22.
Remote/tele-operated driving essential feature of CCAM and AV.

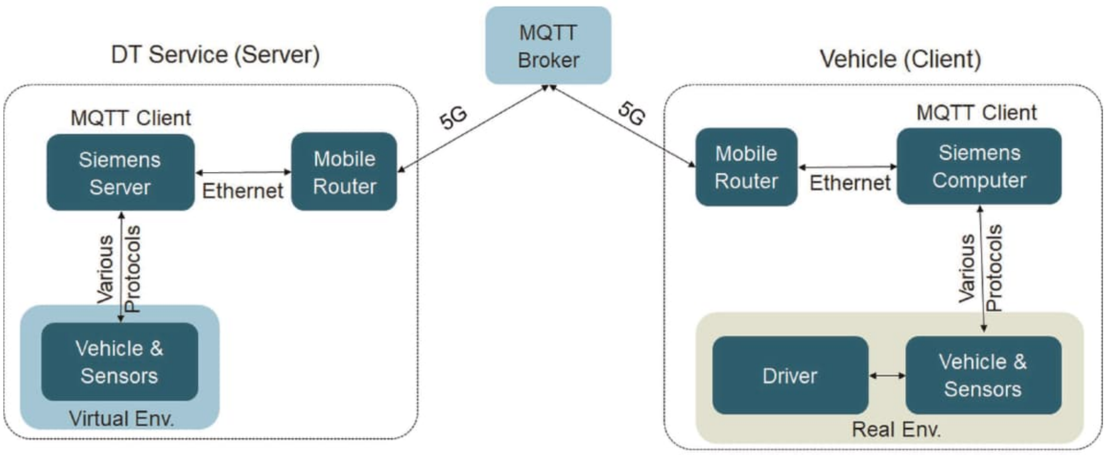


Figure 23.
Mixed-reality testing feature of CCAM.

with data saving options. The vehicle, its surroundings, including the infrastructure are all remotely monitored via the vehicle's onboard sensors see lower-right part of **Figure 21**. This real-time data, in combination with digital twins, can be used by the engineers to assess the vehicle safety/performance and improve/update the software whenever is necessary [3, 24].

The modular setup shown in **Figure 21** can be extended to remote/tele-operated driving, as presented in **Figure 22**. The remote driver perceives the images, GPS data, and provides input (acceleration/braking/steering) values to the remote vehicle *via* the MQTT cloud. The remote driver operates in a virtual environment and the vehicle is in the real-world environment, see **Figure 22**.

Remote monitoring - using advanced and safe communication networks with low latency - is an essential sub-component of the safe teleoperation, where the latter is mainly used during emergency maneuvers of AVs.

The presented applications in this subsection are essential features of CCAM and AVs, so proper testing and suitable test environments are necessary before deployment.

The mixed-reality testing setup shown in **Figure 23** allows testing efficiently and safely different scenarios in combination with V2X (e.g., roadside unit sensors), where virtual V2X objects are created in the virtual environment and can be injected remotely *via* the MQTT cloud, directly into the vehicle CAN bus.

Additionally, the setup allows to testing CCAM applications in the presence of packet loss, communication jitter, and different values of communication latency.

6. Conclusion

This chapter highlighted that the complexity of ADAS/AV requires efficient and robust testing methods to ensure safety and reliability. In this sense Siemens' autonomy workflow and toolchain for testing and virtual validation of ADAS/AV functions was presented.

The introduction section described the growing ADAS and AV market, driven by the integration of advanced technologies and increasing global demand. It addressed the main issues that AV development faces, like guaranteeing scalability, improving testing effectiveness, and getting around infrastructure constraints. The section highlighted virtual validation and scenario-based testing as effective testing of AV systems in a variety of circumstances. Siemens' toolchain was further presented for scenario generation and validation to speed up development.

In Section 2 we focused on the verification and validation workflow starting with describing the ODD, the definition of the DDT, and requirements elicitation according to applicable legislation and standards. In this sense, a detailed overview of the applicable legislation and relevant standards is provided. Furthermore, the requirements management process highlighted the importance of traceability between requirements and test cases/results.

Furthermore, the combination of known-safe scenario extraction and the generation of both known and unknown-unsafe scenarios ensures robust testing of autonomous vehicle systems, addressing both predictable and unforeseen hazards are discussed in Section 3.

The integration of standard scenarios from regulatory and industry standards highlights the features of the toolchain meeting global safety and performance requirements. In addition, advanced techniques in unknown-unsafe scenario creation enable developers to identify critical risks, ensuring improved system safety and reliability in complex and rare real-world scenarios.

In Section 4 we highlighted key components of the Siemens methodology and tooling for efficient, full-scale virtual scenario-based testing, including high-fidelity simulation models, automated scenario selection, advanced scenario sampling methods, and methods for requirements coverage.

It also demonstrated how the methodology and tooling conform with aspects of the simulation credibility assessment frameworks of NATM and EU 2022/1426 and how a solid safety argument based on virtual scenario-based testing can be achieved.

Finally, in Section 5, as an essential part of the verification and validation workflow, SiL and HiL simulation environments were discussed. Especially for HiL simulations, three challenges were presented, being scalability with respect to the number of simulated sensors, automotive-grade interfacing to make the gap toward actual implementation as small as possible, and synchronization of the various simulation components to ensure deterministic execution of the simulation.

It was shown that scalability is obtained by implementing a federated architecture of the simulation, thereby ensuring real-time execution independent of the number of simulated sensors.

In addition, essential CCAM and AV features, such as remote monitoring and tele-operated driving have been presented and a modular/distributed test environment has been described.

Acknowledgements

Co-funded by the European Union 6GSNS

HORIZON JU Innovation Actions - 101139048 - ENVELOPE - HORIZON-JU-SNS-2023. This project has received funding from the European Union's Smart Networks and Services Joint Undertaking (SNS JU) under grant agreement no. 101139048. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or SNS JU. Neither the European Union nor SNS JU can be held responsible for them.

Funded by the European Union NextGenerationEU Digital Infrastructure for Future-proof Mobility

This project has received funding from the European Union in the NextGenerationEU program under grant agreement no. NGFDI2201, Digital Infrastructure for Future-proof Mobility (DITM).

Funded by the European Union NextGenerationEU SUNRISE

This project has received funding from the European Union's Horizon Europe Research and Innovation Actions under grant agreement no.101069573, a safety assurance framework for connected, automated mobility systems (SUNRISE).

Author details

Alexandru Forrai*†, Mohsen Alirezaei†, Tajinder Singh†, Amit Gali† and Jeroen Ploeg†
Siemens Industry Software Netherlands B.V., Helmond, The Netherlands

*Address all correspondence to: alexandru.forrai@siemens.com

† These authors contributed equally.

References

[1] Markets and Markets. Advanced driver assistance systems market report. 2030. Available from: https://www.marketsandmarkets.com/Market-Reports/driver-assistance-systems-market-1201.html [Assessed: January 30, 2025]

[2] McKinsey and Company. Autonomous driving's future: Convenient and connected. Available from: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/autonomous-drivings-future-convenient-and-connected [Assessed: January 30, 2025]

[3] Forrai A, Gali A, Barosan I. Application of digital twins for connected, cooperative and automated mobility. In: IEEE Third Conference on Information Technology and Data Science, Debrecen. New York, USA: IEEE; 2024. pp. 44-50

[4] UNECE. The new assessment/test method for automated driving (NATM). Available from: https://unece.org/sites/default/files/2024-02/ECE_TRANS_WP.29_2022_58e.pdf [Accessed: January 30, 2025]

[5] ISO 26262 Road Vehicles. Functional Safety. 2nd ed. Geneva, Switzerland: ISO; 2018

[6] ISO 21448 Road Vehicles. Safety of the Intended Functionality. 1st ed. Geneva, Switzerland: ISO; 2022

[7] ISO 22737. Low-Speed Automated Driving (LSAD) Systems for Predefined Routes. 1st ed. Geneva, Switzerland: ISO; 2021

[8] BSI PAS 1883. Operational Design Domain (ODD) Taxonomy for an Automated Driving System (ADS) – Specification. 1st ed. London, UK: BSI (British Standards Institution); 2020

[9] Singh T, van Hassel E, Sheorey A, Alirezaei M. A systematic approach for creation of SOTIF's unknown unsafe scenarios: An optimization based method. In: SAE Technical Paper. Pennsylvania, USA: SAE International; 2024

[10] Rajesh N, van Hassel E, Alirezaei M. Unknown-unsafe scenario generation for verification and validation of automated vehicles. In: IEEE 26th International Conference on Intelligent Transportation Systems (ITSC). New York, USA: IEEE; 2023

[11] SIEMENS. Simcenter HEEDS. Available from: https://plm.sw.siemens.com/en-US/simcenter/integration-solutions/heeds/ [Accessed: November 25, 2024]

[12] SIEMENS. Simcenter Prescan. Available from: https://plm.sw.siemens.com/en-US/simcenter/autonomous-vehicle-solutions/prescan/ [Accessed: November 26, 2024]

[13] Association for Standardization of Automation and Measuring Systems. ASAM OpenDrive. Available from: https://www.asam.net/standards/detail/opendrive/ [Accessed: October 26, 2024]

[14] Association for Standardization of Automation and Measuring Systems. ASAM OpenScenario. Available from: https://www.asam.net/standards/detail/openscenario/ [Accessed: October 26, 2024]

[15] European Union. Uniform procedures and technical specifications for the type-approval of the automated driving system (ADS) of fully automated

vehicles. Available from: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32022R1426 [Accessed: October 10, 2024]

[16] International Standards Organization. ISO 34503:2023. Available from: https://www.iso.org/standard/78952.html [Accessed: June 15, 2024]

[17] Association for Standardization of Automation and Measuring Systems. ASAM OpenLabel. Available from: https://www.asam.net/standards/detail/openlabel/ [Accessed: June 18, 2024]

[18] SIEMENS. Simcenter autonomy data analysis. Available from: https://plm.sw.siemens.com/en-US/simcenter/autonomous-vehicle-solutions/autonomy-data-analysis-solution/ [Accessed: November 27, 2024]

[19] Alirezaei M, Forrai A, Tong S, Singh T. Siemens toolchain for testing and virtual validation of ADAS/AV functions. ATZ Magazine Worldwide. 2024;**126**:16-21. DOI: 10.1007/s38311-024-1948-x

[20] Mobileye. SuperVision. Available from: https://www.mobileye.com/solutions/super-vision [Accessed: November 25, 2024]

[21] Baidu Apollo. Apollo open platform. Available from: https://developer.apollo.auto [Accessed: November 25, 2024]

[22] IEEE standard for a precision clock synchronization protocol for networked measurement and control systems. In: IEEE Std 1588–2019 (Revision of IEEE Std 1588-2008), 16 June 2020. New York, USA: IEEE; 2024

[23] NVIDIA. DLSS. Available from: https://developer.nvidia.com/rtx/dlss [Accessed: November 25, 2024]

[24] Abdelsalam M, Forrai A. A design methodology for compositional simulation: The digital-twin interconnect framework. In: 23rd Driving Simulation and Virtual Reality Conference and Exhibition, Strasbourg, France, 2024, Product Solutions. Strasbourg, France: DSC2024; 2024. pp. 1-4