Taxonomy of Scenarios for Systems Engineering Activities

Ali Dayan*, Stephan Reuter, Dr. Carsten Patz
and Prof. Dr. Ing. Raoul Zöllner

Abstract:
Scenarios play an important role in the development and release of automated vehicles (AV), as they form the basis for system development, testing and ensuring safety in various situations, including potentially hazardous ones. However, there are a variety of definitions and applications of scenarios, often with different accuracy requirements. The relationships between the different scenario concepts in use case definition, Hazard Analysis and Risk Assessment (HARA), Safety Analysis and Risk Assessment (SARA), and testing often remains vague. This paper aims to develop a consistent definition and taxonomy of scenarios and assign them to different systems engineering processes.
Keywords: Scenario, Taxonomy, Verification and Validation

1 Introduction

As technology advances and highly automated driving functions are developed, car manufacturers and suppliers face new challenges, particularly in ensuring the safety of these systems [1]. The notion of scenarios is a key concept for developing and testing such complex systems. This in particular includes the testing of automated driving systems across a wide range of conditions, from routine driving situations to potentially hazardous or extreme events but is not at all limited to that [2].

Despite their importance, the definition and application of scenarios in the automotive industry are not standardized [3]. Various standards and methodologies interpret and use the terms differently, leading to inconsistencies and misunderstandings. This lack of standardization poses a potential risk, particularly in safety-critical areas, as ambiguities in the scenario usage can expose gaps in a safety argumentation. Consequently, the need for a consistent, comprehensive definition and a structured taxonomy of scenarios is becoming increasingly urgent.

This paper addresses the challenge of developing a clear and consistent definition of scenarios and establishing a structured taxonomy that integrates them into different systems engineering activities. The authors propose a standardized methodology for creating and applying scenarios in various stages of the V-Model. Guidelines and examples are provided to demonstrate the appropriate abstraction levels of scenarios for specific systems engineering activities.

* Ali Dayan, Valeo, Laiernstr. 12, 74321 Bietigheim-Bissingen, ali.dayan@valeo.com
Stephan Reuter, Valeo, Laiernstr. 12, 74321 Bietigheim-Bissingen, stephan.reuter@valeo.com
Dr. Carsten Patz, Valeo, Laiernstr. 12, 74321 Bietigheim-Bissingen, carsten.patz@valeo.com
Prof. Dr. Ing. Raoul Zöllner, Hochschule Heilbronn, 74081 Heilbronn, raoul.zoellner@hs-heilbronn.de

2 Definitions and Concepts of Scenarios in Research and Standardization

This Chapter explores the multifaceted concept of "scenario" within the context of automated vehicles (AV). It examines diverse definitions, relevant standards (ISO 34501 [6], ISO 34504 [7], ISO 21448 [8], ISO 26262 [9]), and practical applications in initiatives such as PEGASUS [11], and ASAM OpenODD [5]. PEGASUS is an initiative that focuses on defining standards for the testing and validation of automated driving systems to ensure safety. ASAM OpenODD provides a standardized format for the description of Operational Design Domains (ODDs) that define the conditions for the operation of these systems. Together they address critical aspects of the development and deployment of automated driving systems. At the end of Chapter 2 , we present our own definition of scenarios, derived from various existing definitions.

Scene

A scene describes a static snapshot of the environment, encompassing scenery, dynamic elements, actors, their self-representations, and the relationships between these entities. In a simulated world, scenes can be fully objective (ground truth), whereas in the real world, they are incomplete, potentially inaccurate, uncertain, and viewed from one or several subjective perspectives [4].

Situation

A situation comprises all relevant circumstances at a specific moment that influence the selection of an appropriate behavior pattern. It includes conditions, options, and influences for behavior and is derived from a scene through processes of information selection and augmentation, based on both transient (mission-specific) and permanent goals and values. Thus, situations are inherently subjective, reflecting the perspective of an element [4].

Scenario

Definition 1: A scenario describes the temporal evolution from one scene to another in a sequence. It starts with an initial scene and may include specified actions, events, goals, and values to outline this temporal development. Unlike a scene, a scenario covers a span of time [4].

Definition 2: A scenario is a description of how the world changes over time, often from a specific viewpoint. Within the vehicle and driving context, this includes changes in both static (e.g., road layout, road furniture) and dynamic (e.g., weather, lighting, moving objects, people, and traffic signals) elements of the environment, regardless of whether the environment is simulated, real, or a mix of both [6].

Summary of Definitions:

The two definitions differ in their focus: the first emphasizes the sequential and structural aspects of scenarios, while the second takes a broader, contextual perspective that recognizes the multilayered evolution of environmental elements over time. These differences illustrate the different interpretations of scenarios in the literature.

Relevant ISO standards

ISO 34501:2022 - Test scenarios for automated driving systems - Vocabulary: Offers a structured methodology for defining, classifying and deriving scenarios that are relevant for the development of automated driving systems. It includes a taxonomy and techniques for deriving scenarios from real traffic situations [6].

ISO 34504:2024 - Test scenarios for automated driving systems - Scenario categorization: Offers a methodology for categorizing scenarios which are relevant for the safety assessment. It also contains a taxonomy for classifying scenarios based on environmental conditions and traffic situations that support risk identification and assessment, including rare and critical cases [7].

ISO 21448 - Safety of the Intended Functionality - SOTIF: Goes beyond traditional functional safety by addressing risks that arise from the intended functionality. It focuses on scenarios in which no malfunction of the system occurs, but risks may arise due to inadequate specifications or performance, and aims to mitigate such risks through scenario analysis [8].

ISO 26262 - Road vehicles - Functional safety: Focuses on the functional safety of electrical and electronic systems in vehicles. Scenarios are used to identify and assess hazards and risks associated with system failures. The Hazard Analysis and Risk Assessment (HARA) process in particular uses scenarios to identify hazards and define suitable safety measures [9].

Key differences and contributions of ISO standards:

Purpose and focus:

- ISO 34501 addresses scenario development.
- ISO 34504 focuses on scenario categorization and classification.
- ISO 21448 focuses on risks from the intended functionality.
- ISO 26262 emphasizes preventing risks from system malfunctions.

Methodology:

- ISO 34501 introduces a systematic approach to scenario management.
- ISO 34504 defines criteria for the classification and organization of scenarios.
- ISO 21448 concentrates on scenarios without system failures.
- ISO 26262 focuses on risk assessment from failures.

Application: These standards address various safety aspects at different stages of AVs development and validation and ensure comprehensive safety consideration.

Conclusion: This Chapter highlights the central role of scenarios in the development and validation of AV's. By analyzing key definitions, standards and initiatives, the different interpretations and applications of scenarios are highlighted. Together, the ISO standards discussed provide a robust framework for scenario development at different stages of AV system design. At the end of Chapter 2, we propose a definition of scenarios in the context of AV, developed from a synthesis of the various existing definitions. This definition captures the dynamic sequence of scenes that evolve over time and includes both static and dynamic elements of the environment from the system's perspective.

3 Scenario abstraction level

This Chapter examines the critical role of scenario abstraction levels in AV development, focusing on the PEGASUS [11] project's three level approach (functional, logical and concrete). The Chapter compares this approach with the ASAM OpenODD [5] standard to highlight the respective strengths, weaknesses and potential synergies. It then analyzes the benefits and challenges of each level of abstraction and explains why the PEGASUS framework was chosen for the systems engineering activities described in this paper.

Figure 1 shows the different scenario abstraction levels from the PEGASUS project. A difference is made between three different levels, which are subsequently used to describe the scenarios in the various systems engineering activities and form the basis of this work.

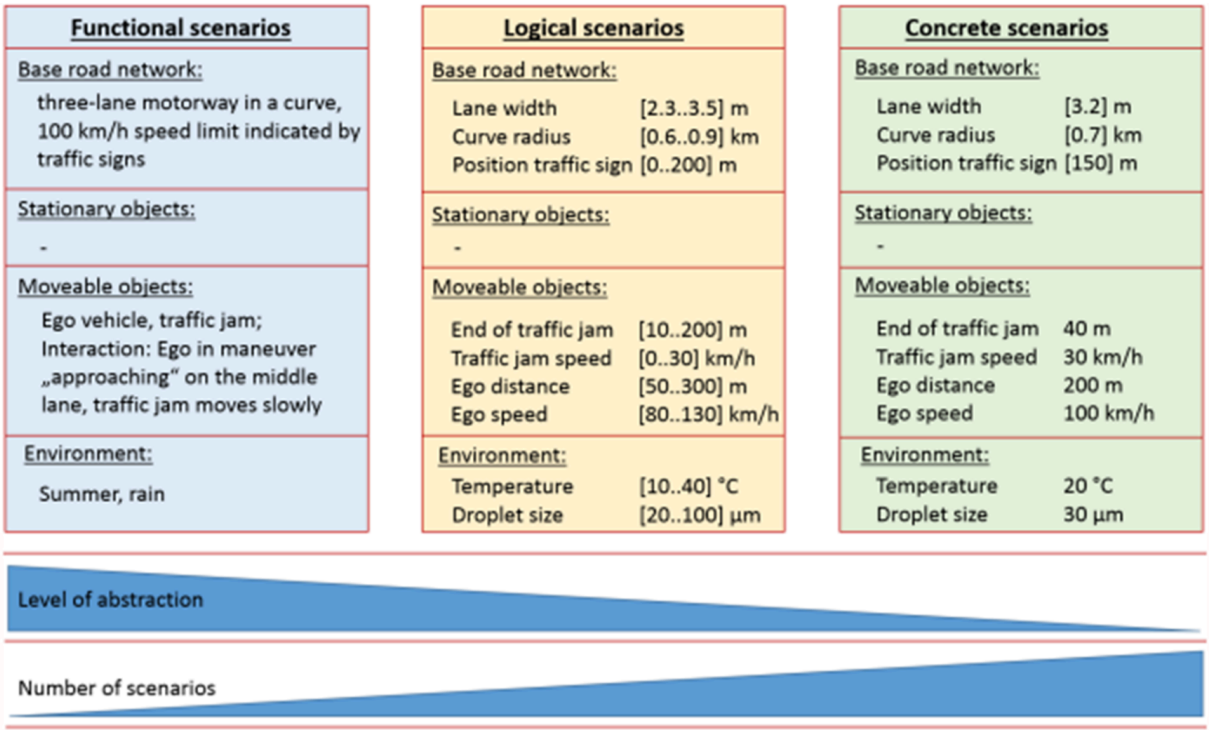

Figure 1: Scenario abstraction levels from the PEGASUS project [10]

Functional scenarios – PEGASUS:
Functional scenarios represent operations on a seminatural level and describe domain entities and their relationships using a language representation. These scenarios are consistent in themselves, whereby the vocabulary is adapted to the respective use case and domain [10].

Logical scenarios – PEGASUS:
Logical scenarios define operations at the level of a state space in which entities and their relationships are represented by parameter ranges. These ranges can include probability distributions, correlations or numerical conditions. Logical scenarios are formally notated and contain all elements necessary to derive technical requirements for system implementation [10].

Concrete scenarios – PEGASUS:
Concrete scenarios clearly represent operational scenarios at the state space level. They represent entities and the relationships between these entities using concrete values for each parameter in the state space [10].

Abstract Scenarios - ASAM OpenODD:
Abstract scenarios provide a conceptualization of scenarios at the level of scenario intention. They provide a formal, declarative description that is both machine- and human-readable. These scenarios capture dependencies and relationships between attributes and behaviors using constraints. In this way, several concrete scenarios can be created from a single abstract description [5].

Logical Scenarios - ASAM OpenODD:
Similar to the PEGASUS project, the logical scenarios in ASAM OpenODD include parameter ranges that can contain distributions or correlations. These scenarios are specifically designed for compatibility with simulation environments and are widely used in the development of automated driving systems [5].

Concrete Scenarios - ASAM OpenODD:
Concrete scenarios are scenarios in which all parameters are set to specific values. This can be done by specifying concrete locations and trajectories or by referring to deterministic models. ASAM OpenODD's approach to concrete scenarios ensures that they can be used directly in simulations or real tests, emphasizing the transition from the abstract to the concrete for practical application [5].

Comparison:
Abstraction vs. Concretization: Both frameworks use abstraction to generalize scenarios and concretization to specify them for testing or simulation. However, ASAM OpenODD introduces a more structured approach to abstraction levels, with a focus on machine readability and simulation compatibility.

Use in Development: The PEGASUS abstraction levels are primarily used in the context of system development activities, while the ASAM OpenODD scenarios are designed for the broader development and validation of automated driving systems, including the usage in simulation tools.

Flexibility and Reuse: The abstract scenarios of ASAM OpenODD offer a greater flexibility in the creation of scenarios, so that one scenario description can generate numerous test cases. This promotes efficiency in achieving test coverage.

This comparison helps to understand how these different frameworks complement each other or can be used at various stages of the AV development and validation process. In this paper, we adopt the PEGASUS abstraction levels to create scenarios for the systems engineering activities.

Advantages and challenges of the scenario abstraction levels
Functional Scenarios:
Advantages: Early conceptualization, domain-specific language, flexibility.
Challenges: Lack of clarity, transition to implementation.
Logical Scenarios:
Advantages: Formalization, parameterization, simulation.
Challenges: Complexity, misinterpretation.
Concrete Scenarios:
Advantages: Realism, validation, reproducibility.
Challenges: Data effort, scalability, generalizability.

4 Scenarios in Systems Engineering Activities

This Chapter illustrates how scenarios are practically applied throughout the systems engineering V-cycle for AVs. Using two examples, parking between two cars (with a pedestrian) and highway lane changing (with a broken-down vehicle in front of the AV). This Chapter demonstrates how scenario granularity and abstraction levels, based on the PEGASUS [11] framework's six layers [11], vary across different systems engineering activities. Additionally, this Chapter analyzes the connections between scenarios across the V-cycle, highlighting the underlying taxonomy and showing how functional scenarios inform the creation of concrete test cases.

Therefore, Figure 2 is showing the various scenario abstraction levels used for different activities. This approach integrates scenarios into the Systems Engineering activities, enhancing the understanding, development, testing, and communication of system capabilities and limitations, thereby supporting a more robust, safe, and effective system design and operation.

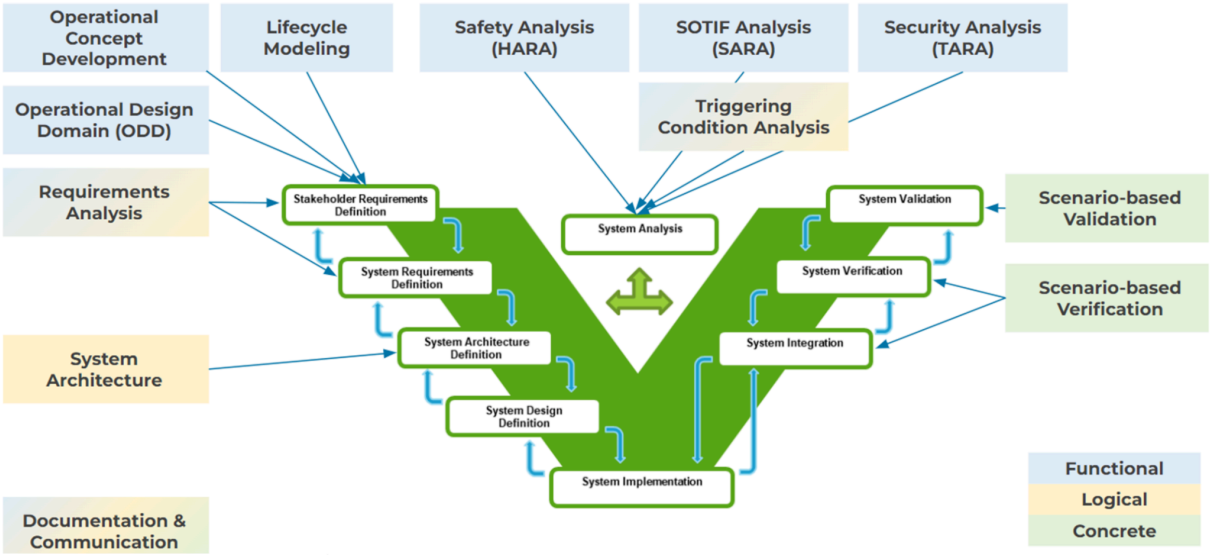

Figure 2: Scenarios along the V-Cycle activities

The core message of this paper is that we define functional, logical, and concrete scenarios at different stages of the development process. Specifically, we use **functional scenarios** for HARA, SARA (Safety Analysis and Risk

Assessment), TARA (Threat Analysis and Risk Assessment), ODD (Operational Design Domain), and Stakeholder Requirements analysis, **logical scenarios** for System Requirements and Triggering Conditions, and **concrete scenarios** for Testing (Verification & Validation). This structured approach ensures a systematic progression from conceptualization to detailed validation.

By employing functional scenarios in the early development stages (HARA, SARA, TARA, ODD, and Stakeholder Requirements), we achieve several key advantages:

- **Breadth of Analysis:** Functional scenarios enable a comprehensive investigation of potential hazards, risks, threats, operating conditions and desired functionalities that are not limited by specific implementation details.
- **Analytical Efficiency:** Abstraction reduces complexity by limiting the number of scenarios considered, thus enabling efficient analysis prior to detailed design.
- **Emphasis on Functionality:** Functional scenarios give priority to functional behavior and reactions at system level over specific implementations. For stakeholder requirements, this means that they focus on what the system should achieve, not how.
- **Adaptability and Reusability:** Functional scenarios are adaptable to design changes and reusable across different implementations, ensuring continuous relevance during development.
- **Scalability to Concrete Scenarios:** These functional scenarios form the basis for deriving more concrete, detailed scenarios in later phases, such as testing and validation, and for refining functional stakeholder requirements into concrete system requirements.

This approach ensures that stakeholders can clearly express their needs without imposing too many restrictions on design options, thus promoting a robust and adaptable development process.

Next, we use **logical scenarios** to define **System Requirements** and **Triggering Conditions**, offering several key advantages:

- **System Requirements:** Formulating system requirements in the form of logical scenarios increases clarity and testability, as they describe specific system behavior in defined contexts. For example, the statement “If a pedestrian enters the vehicle's path, the system must detect the pedestrian and initiate braking” is more concrete and testable than the simple statement “The system must detect pedestrians”. This method also improves completeness by taking into account different operational contexts and facilitating traceability between higher level objectives and specific functionalities.
- **Triggering Conditions:** Logical scenarios are well suited for defining trigger conditions due to their inherent context and precision. They ensure that functions are only activated under appropriate circumstances, using logical and relational operators for precise definitions.

Logical scenarios provide a good balance between specificity and flexibility. They are more detailed than functional scenarios and clarify what the system should do and when, while avoiding overspecification of concrete scenarios and saving implementation details for later design phases.

Finally, we define **concrete scenarios** for **Testing (V&V)**, because the aim of this phase is to precisely evaluate system performance using specific, quantifiable metrics. Concrete scenarios provide the necessary level of detail for reproducible and repeatable tests. These scenarios specify precise parameter values for all relevant elements, such as:

- Ego vehicle: Starting position, velocity and trajectory.
- Other objects (e.g. vehicles, pedestrians): Detailed trajectories, velocities.
- Environment: Specific road geometry, lighting conditions, weather.

This concreteness ensures that the tests cover precisely the defined situations and that the results can be compared across different test runs and system versions. This precision is essential for the objective evaluation of system performance within the defined ODD and for the fulfillment of safety and validation requirements.

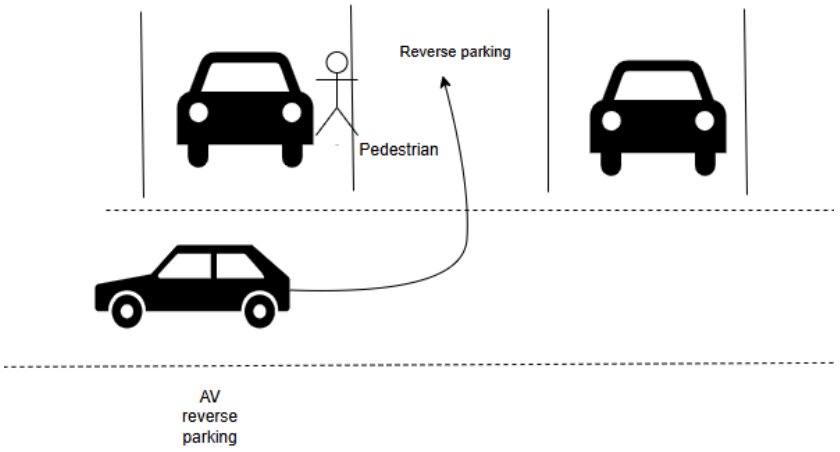

Figure 3: Reverse parking of AV between two cars with a pedestrian

Figure 3 illustrates a reverse parking scenario between two vehicles, with a pedestrian standing left to the parking slot. This scenario shall feature an available parking space, clearly marked road lines, a pedestrian and favorable weather conditions, ensuring clear visibility.

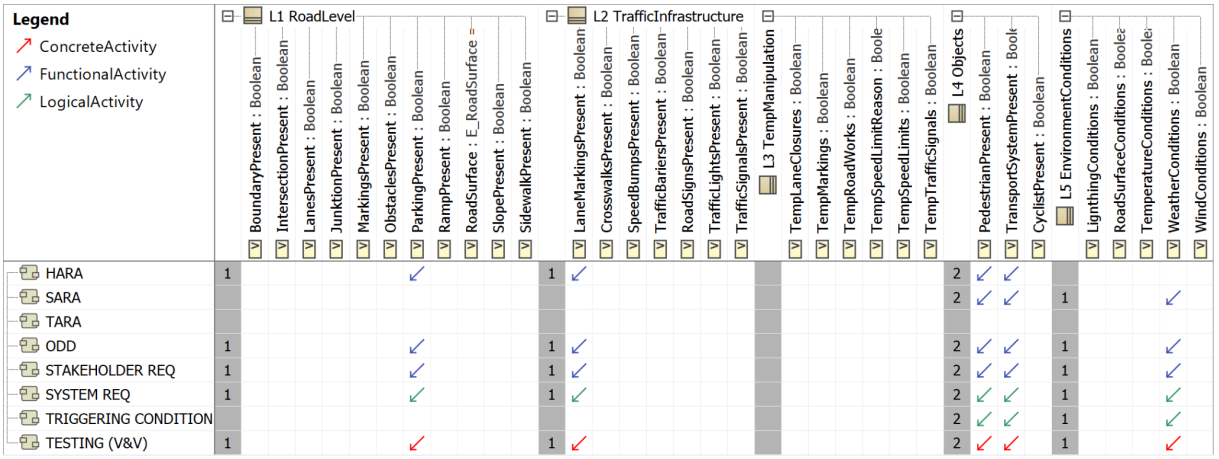

Figure 4: Granularity of scenarios for parking scenario

Figure 4 depicts the granularity of scenarios applied across various V-cycle activities (shown in the rows on the left). Positioned alongside are the five layers of the PEGASUS [11] model, each accompanied by its own distinct data model.

HARA: Focuses on identifying potential hazards during reverse parking. L1 and L2 are crucial for collision avoidance. L4 (pedestrian presence) is critical as a dynamic hazard.
SARA: Focuses on how the system deals with challenges that may affect its perception, decisionmaking and overall functionality that can be affected by L5 (e.g. weather) and L3 (e.g. temporary manipulations that affect visibility). The pedestrian (L4) is a critical obstacle. L1 and L2 are generally considered, but the focus here is on perception and reaction to the environment and dynamic objects.
ODD: Defines the conditions under which the parking system is designed to operate safely. This includes L1 (Road Level) L2 (Traffic infrastructure, parking space markings) and L5 (environmental conditions the system is designed to handle).
STAKEHOLDER REQ: Express the stakeholder expectations for the system, such as parking in well-marked spaces (L2) and functioning in various weather conditions (L5). These requirements drive the definition of the ODD.
SYSTEM REQ: Define the functional requirements enabling the system to meet the stakeholder needs. These requirements cover responses to Road Level parameters (L1), Traffic infrastructure (L2), Temporary Manipulation of L1 and L2 (L3), object detection and avoidance (L4), and adaptation to broader environment and lighting conditions (L5). Level 3 is not explicitly stated in the figure above.
TRIGGERING CONDITIONS: Specific conditions of a driving scenario that serve as an initiator for a subsequent system reaction, possibly leading to a hazardous event. This directly affects L4 (position and movement of the pedestrian). L3 (temporary limitation of visibility e.g. fog, smoke) and L5 (lightning conditions) influence the effectiveness of pedestrian detection and therefore the trigger conditions.
V&V: Specifies test cases covering all relevant layers. This includes precise measurements for the parking space (L1), lane marking positions (L2), lighting and weather (L3 and L5), and precise pedestrian positions and movements, as well as parked vehicles (L4). V&V activities should confirm that the system requirements are met throughout the ODD.

Figure 5 illustrates the relationship between the scenario defined in HARA and the scenario applied in the test phase (V&V). It illustrates how a functional scenario defined in earlier project phases such as HARA is concretized in the test phase by a specific test case. This example illustrates the underlying taxonomy.

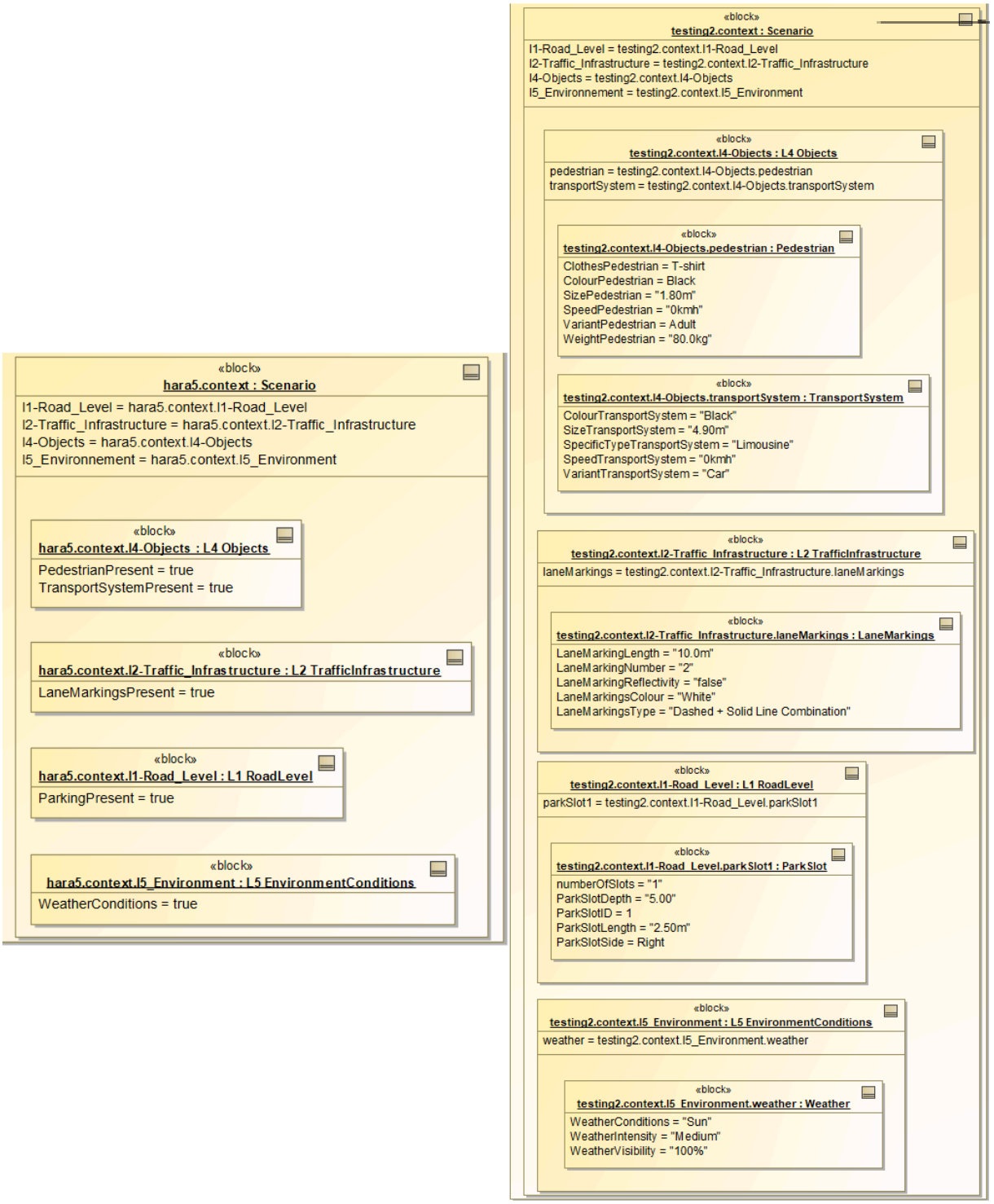

Figure 5: Relation between scenarios in HARA and Testing

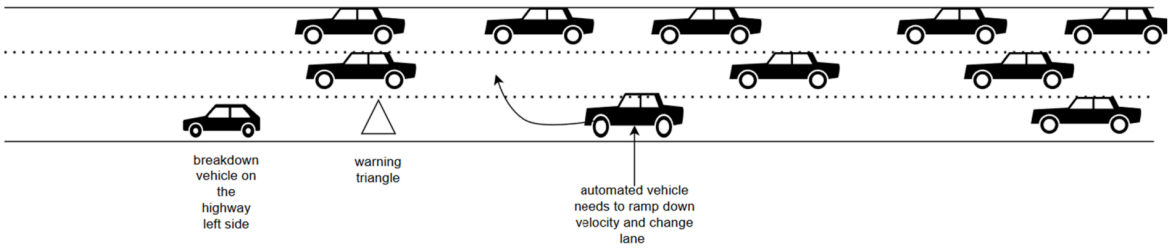

Figure 6: Lane changing of AV

Figure 6 shows the second example of a highway scenario in which an AV driving in the left lane approaches a blocked vehicle. The scenario requires the AV to safely reduce its speed and change lanes.

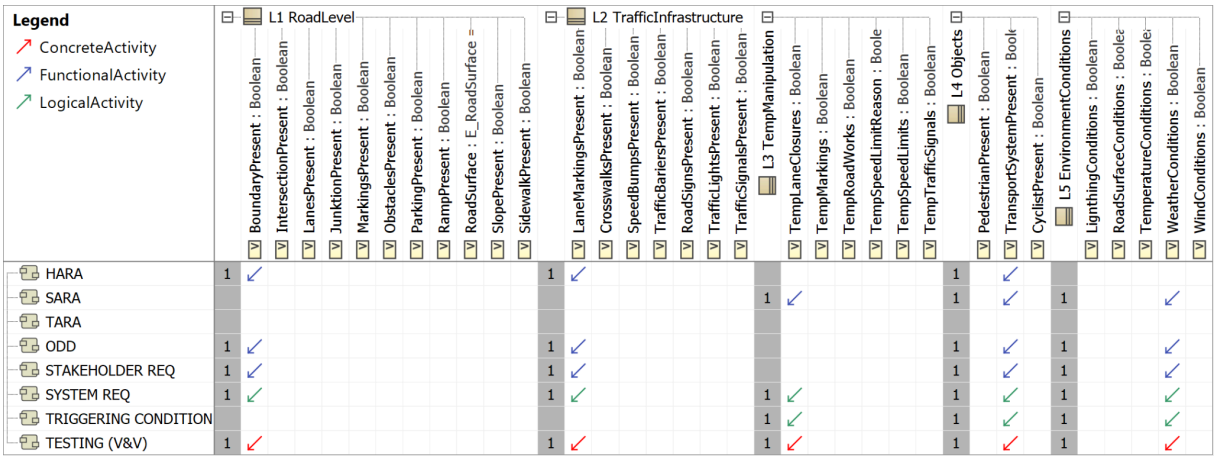

Figure 7: Granularity of scenarios for highway example

Figure 7 illustrates the granularity of the scenarios used in different V-cycle activities for the highway example. In this scenario, L3 (Temporary Manipulation) affects L1 and L2, represented by the stalled vehicle acting as a temporary lane closure. From the perspective of ODD, Stakeholder Requirements and HARA, no scenario description is required for L3. For SARA, a functional scenario description is used as the stalled vehicle could cause a distraction. The system requirements and trigger conditions are defined as logical scenarios to account for the system's response to transient changes. As in the previous phases, testing is based on concrete scenario descriptions.

Figure 8 shows the relationship between the scenarios in the system requirements and the scenarios used during testing (V&V). It shows how a logical scenario, which was defined in earlier phases such as the system requirements, is converted into a concrete test case in the test phase. This example illustrates the basic taxonomy and the transition from logical to concrete scenarios.

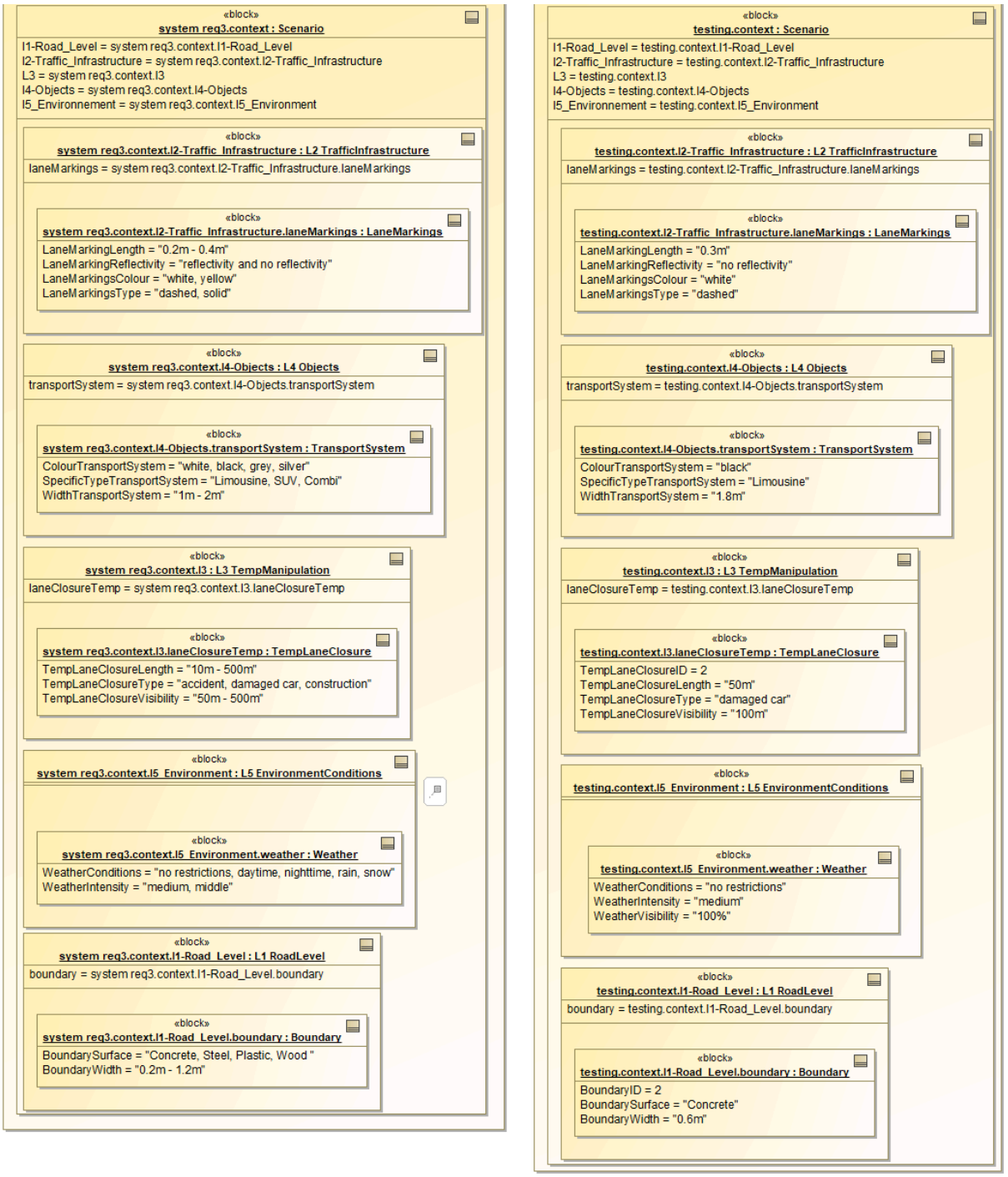

Figure 8: Relation between scenarios in System Requirements and Testing (V&V)

5 Conclusion

This paper explored the central role of scenarios in the development and validation of ADAS / ADS. The authors demonstrated that scenarios are not only essential for the verification and validation of systems, but also play a crucial role in various activities of the Systems Engineering process, such as performing a HARA, SARA, or to define the ODD. By comparing standards such as ISO 34501, ISO 34504, ISO 21448 and ISO 26262 as well as projects such as PEGASUS and ASAM OpenODD, a consistent and efficient application of scenarios in the context of the safety and

functionality of autonomous vehicles was defined on different existing standards. Ultimately, the PEGASUS approach was used and the scenario abstraction levels were adapted to the V-Cycle activities.

The granularity of scenarios varies from functional descriptions in the early stages of development to concrete, detailed scenarios for the testing phases. This approach enables the systematic addressing of safety-critical aspects and ensures that the systems can operate both safely and functionally in real, dynamic environments.

In conclusion, the methodology of scenarios forms an indispensable foundation for the development and validation of autonomous vehicles, bridging the gap between theoretical requirements and practical implementation, thereby contributing to the fulfillment of high safety and functional standards.

References

[1] Menzel, T., Bagschik, G., & Maurer, M. (2018). Scenarios for Development, Test and Validation of Automated Vehicles.

[2] Zhang, X., et al. (2021). Finding Critical Scenarios for Automated Driving Systems: A Systematic Literature Review.

[3] Li, X. (2020). A Scenario-Based Development Framework for Autonomous Driving.

[4] Ulbrich, S., Menzel, T., Reschka, A., Schuldt, F., & Maurer, M. (2015). Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving.

[5] ASAM e.V. (2021). OpenODD: A Standard for Operational Design Domains. [Online]. Available: https://www.asam.net/index.php?eID=dumpFile&t=f&f=4544&token=1260ce1c4f0afdbe18261f7137c689b1d9c27576

[6] ISO 34501:2022. Road Vehicles - Test Scenarios for Automated Driving Systems - Vocabulary. [Online]. Available: https://www.iso.org/obp/ui/en/#iso:std:iso:34501:ed-1:v1:en

[7] ISO 34504:2024. Road Vehicles - Test Scenarios for Automated Driving Systems - Scenario categorization.[Online]. Available: https://www.iso.org/obp/ui/es/#iso:std:iso:34504:ed-1:v1:en

[8] ISO 21448:2022. Road Vehicles - Safety of the Intended Functionality. [Online]. Available: https://www.iso.org/obp/ui/en/#iso:std:iso:21448:ed-1:v1:en

[9] ISO 26262:2018. Road Vehicles - Functional Safety. [Online]. Available: https://www.iso.org/obp/ui/en/#iso:std:iso:26262:-1:ed-2:v1:en

[10] PEGASUS Projekt. Szenarienbeschreibung. [Online]. Available: https://www.pegasusprojekt.de/files/tmpl/PDF-Symposium/04_Scenario-Description.pdf

[11] PEGASUS Projekt. Gesamtmethode. [Online]. Available: https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/PEGASUS-Gesamtmethode.pdf