SYSTEMATIC APPROACH TOWARDS SAFETY OF THE INTENDED FUNCTIONALITY: ELICITING AND EVALUATING DIFFICULT ENVIRONMENTAL CONDITIONS FOR AUTOMATED DRIVING SYSTEMS

Dissertation

zur Erlangung des Grades eines

D o k t o r s d e r I n g e n i e u r w i s s e n s c h a f t e n

der Technischen Universität Dortmund
an der Fakultät für Informatik

von

Frau Zhijing Zhu

Dortmund

6. Oktober 2025

Tag der mündlichen Prüfung:	20.08.2025
Dekan:	Prof. Dr. Jens Teubner
Erstgutachter:	Prof. Dr. rer. nat. Falk Howar
Zweitgutachter:	Prof. Dr. Martin Törngren

Kurzfassung

Mit dem Übergang von Fahrerassistenzsystemen zu automatisierten Fahrzeugen wird Sicherheit zu einem zentralen Ziel für die breite Markteinführung. Die funktionale Sicherheit im niedrigen Automatisierungsbereich (L0–L2 gemäß SAE-Standard) ist mithilfe der in der ISO 26262 beschriebenen Methoden gut messbar und beherrschbar. Da jedoch der Mensch zunehmend als Rückfallebene entfällt, reicht ISO 26262 für die Analyse bestimmter kritischer Situationen nicht mehr aus. In solchen Fällen resultieren Fehler nicht nur aus dem E/E-System des Fahrzeugs, sondern auch aus schwierigen Umweltbedingungen, die aufgrund von Spezifikations- oder Funktionsunzulänglichkeiten möglicherweise unzureichend behandelt werden. Diese Umweltbedingungen sind entscheidend für die Absicherung, da szenarienbasiertes Testen auf ihrer Grundlage effizienter und praktikabler ist als eine vollständige Erkundung des Szenarienraums. Diese Arbeit stellt einen systematischen Ansatz vor, um die Herausforderungen schwieriger Umweltbedingungen für automatisierte Fahrsysteme (ADS) zu bewältigen. Zunächst wird der Begriff der schwierigen Umweltbedingungen auf Basis des aktuellen Forschungsstands interpretiert. Anschließend werden drei Typen solcher Bedingungen identifiziert: das Vorhandensein oder Fehlen spezifischer Umweltfaktoren im gegebenen Betriebsbereich (ODD), spezifische Verhaltensweisen dieser Faktoren sowie deren Interaktionen. Für jede dieser Typen werden formale, maschinenlesbare Beschreibungen vorgeschlagen, um eine einheitliche Darstellung und Bewertung sowie die Ableitung von Testfällen zu ermöglichen. Zur systematischen Identifikation schwieriger Umweltbedingungen werden sowohl analytische als auch datengetriebene Methoden entwickelt. Eine davon ist die szenarienbasierte Gefährdungs- und Fehleranalyse (Scenario-based Hazard and Fault Analysis, SHFA), die Expert:innen bei der Identifikation solcher Bedingungen durch Analyse potenzieller Gefährdungen unterstützt. Ergänzend wird eine vollautomatische Pipeline zur Rekonstruktion kritischer Fahrszenarien aus realen Testfahrten entwickelt, um schwierige Bedingungen datenbasiert zu identifizieren. Abschließend wird eine umfassende Teststrategie inklusive einer Methode zur Generierung entsprechender Testfälle vorgestellt, um schwierige Umweltbedingungen in szenarienbasiertes Testen zu integrieren. Die Arbeit entstand in enger Zusammenarbeit mit der Industrie und orientiert sich an geltenden Normen wie ISO 21448 und Vorschriften wie EU 2022/1426. Nach aktuellem Kenntnisstand stellt sie das erste konsistente Framework zur Identifikation, Handhabung und Prüfung schwieriger Umweltbedingungen auf Systemebene für ADS dar. Die empirischen Ergebnisse zeigen, dass die vorgestellten Konzepte einen signifikanten Beitrag zur Entwicklung kritischer Testszenarien und zur Weiterentwicklung der szenarienbasierten Verifikation leisten können.

Abstract

With the transition from advanced driver assistance systems to automated vehicles, safety is becoming a key goal for broad market introduction. Functional safety for low-level automated driving (L0-L2 by SAE standard) is well-measurable and manageable based on the methods described by the standard ISO 26262. However, since the fallback of the human driver is gradually taken out of the loop for automated driving systems (ADS), ISO 26262 is insufficient to cover the analysis of certain critical situations. In these situations, failures are not only due to the vehicle's E/E system, but also will be in addition due to difficult environmental situations. They are deemed difficult for ADS, as they could potentially be improperly handled due to certain specifications or design insufficiencies. Such conditions are crucial to safety verification: Organizing scenario-based testing based on them is more efficient and feasible than exhaustively exploring the scenario space. Meanwhile, this requires systematically identifying these conditions within a given Operational Design Domain (ODD) and developing a corresponding test strategy. Thus, this thesis elaborates on a systematic approach to tackle the challenges around difficult environmental conditions for ADS. Firstly, we interpret the nature of difficult conditions based on the state-of-the-art literature. Next, we summarize three types of difficult conditions, namely the presence/absence of specific environmental factors within the given ODD, specific behaviors of environmental factors, and specific interactions among environmental factors. Correspondingly, we propose formal, machine-readable formulations for each type. Consequently, the difficult conditions can be described uniformly, in favor of evaluating these conditions against certain criteria, creating test cases, and tracing test results. After that, we design both analytical and data-driven approaches to systematically identify difficult environmental conditions from the given ODD. On the one hand, we design an analytical method called Scenario-based Hazard and Fault Analysis (SHFA), which supports domain experts to elicit difficult environmental conditions by analyzing potential hazards in driving scenarios with their domain experience. On the other hand, we aim at finding critical scenarios containing difficult environmental conditions from driving data. To that end, we develop a fully automatic pipeline for reconstructing automated vehicle disengagement scenarios from real test drives. Finally, we present an overall test strategy and a test case generation method to integrate difficult conditions into scenario-based testing. This thesis has been developed in close collaboration with industrial automated vehicle production, and therefore, the presented concept and methods target conformance and compliance with the state-of-the-art automotive safety standards like ISO 21448 and regulations like EU 2022/1426. To the best of our knowledge, this thesis provides the first coherent framework for the identification, management, and testing of difficult environmental conditions for verifying ADS on the system level. The empirical findings suggest that concepts and methods around difficult environmental conditions can significantly contribute to identifying and constructing critical test cases, thereby advancing scenario-based verification for automated vehicles.

Disclaimer

The results, opinions and conclusions expressed in this thesis are not necessarily those of Volkswagen Aktiengesellschaft.

Die Ergebnisse, Meinungen und Schlüsse dieser Dissertation sind nicht notwendigerweise die der Volkswagen Aktiengesellschaft.

List of Scientific Contributions

Contributions for this Thesis (Main Author)

I **"Systematization and Identification of Triggering Conditions: A Preliminary Step for Efficient Testing of Autonomous Vehicles"** by Zhijing Zhu, Robin Philipp, Constanze Hungar, and Falk Howar. In: *IEEE Intelligent Vehicles Symposium (IV), 2022, pp. 798-805.* [132]

Comment on participation: This paper interprets the concept of *triggering conditions* from the standard ISO 21448 (related to the term *difficult environmental conditions* in this thesis) and proposes preliminary thoughts to formulate and identify these conditions. I refined the term definitions and explained the relation among triggering conditions, scenario context, hazardous event, and harm in the context of test case design. I designed the preliminary formulation and identification method to elicit triggering conditions. I am the author of all sections. The co-authors contributed mainly to discussing the concepts and were involved in the paper review.

II **"Automatic Disengagement Scenario Reconstruction Based on Urban Test Drives of Automated Vehicles"** by Zhijing Zhu, Robin Philipp, Yongqi Zhao, Constanze Hungar, Jürgen Pannek, and Falk Howar. In: *IEEE Intelligent Vehicles Symposium (IV), 2023, pp. 1-8.* [134]

Comment on participation: This paper introduces an automatic pipeline for reconstructing critical driving scenarios from automated vehicle disengagement records. I prepared the disengagement records from the automated vehicle testing data, which was originally collected by Volkswagen Group. I proposed the original concept of restoring disengagement scenarios from perception object lists. I designed the data processing procedures, which form the pipeline finally. I am the author of all sections. Robin Philipp and Yongqi Zhao contributed to implementing the concept. Constanze Hungar, Jürgen Pannek, and Falk Howar reviewed the paper.

III **"Identifying Difficult Environmental Conditions with Scenario-based Hazard and Fault Analysis"** by Zhijing Zhu, Robin Philipp, Constanze Hungar, and Falk Howar. In: *Springer Nature Computer Safety, Reliability, and Security. SAFECOMP Workshops SASSUR, 2024, 134–147.* [133]

Comment on participation: This paper introduces a systematic method and its application, named SHFA, for the identification of difficult environmental conditions for ADS. I designed the SHFA method and organized a series of workshops with domain experts to apply it. I documented and analyzed the identification results, including formalizing and clustering them. I am the author of all sections. Robin Philipp and Constanze Hungar participated in the workshop organization and the paper review. Falk Howar was involved in the paper review.

IV **"Leveraging Triggering Conditions for Efficient Scenario-Based Testing of Automated Vehicles"** by Zhijing Zhu, Robin Philipp, and Falk Howar. In: *SAE International Journal of Connected and Automated Vehicles, 2025, Volume 8, Number 4* [131]

Comment on participation: This paper discusses the test strategy of triggering conditions in accordance with the objectives in the standard ISO 21448. Besides, it introduces a systematic approach to introduce triggering conditions into scenario-based testing. I analyzed the triggering conditions related requirements in ISO 21448 and interpreted the corresponding verification and validation process. I designed the method to derive test scenarios for triggering conditions, including their parameterization. I designed a simulation-based case study to illustrate the overall concept. I am the author of all sections. The co-authors were involved in the paper review.

Other Contributions (Co-Author)

- **"Automated 3D Object Reference Generation for the Evaluation of Autonomous Vehicle Perception"** by Robin Philipp, Zhijing Zhu, Julian Fuchs, Lukas Hartjen, Fabian Schuldt, and Falk Howar. In: *5th International Conference on System Reliability and Safety (ICSRS), Palermo, Italy, 2021, pp. 312-321.* [83]

  Comment on participation: This paper proposes an offline method to automatically derive dimension and classification labels for 3D objects perceived by ADS. Robin Philipp designed the overall concept of the approach and conducted and evaluated the experiments. I participated in the concept implementation, the discussion of the experiment design, and the result evaluation. Lukas Hartjen, Fabian Schuldt, and Falk Howar were involved in the concept discussion and the paper review.

- **"Systematization of Relevant Road Users for the Evaluation of Autonomous Vehicle Perception"** by Robin Philipp, Jana Rehbein, Felix Grün, Lukas Hartjen, Zhijing Zhu, Fabian Schuldt, and Falk Howar. In: *IEEE International Systems Conference (SysCon), Montreal, Canada, 2022, pp. 1-8.* [82]

  Comment on participation: This paper proposes concepts to determine relevant areas and relevant road users of ADS for a more efficient evaluation of the perception precision. Robin Philipp proposed the overall concept of the approach, designed the six classes of traffic participants and mapped them to driving maneuvers, and implemented the construction of relevant areas. Robin Philipp and Jana Rehbein conducted and evaluated the experiments. Felix Grün, Lukas Hartjen, and I discussed the concept and experiment design and participated in the paper review. Fabian Schuldt and Falk Howar were involved in the paper review.

Acknowledgement

Pursuing a PhD research about automated driving in an industrial environment is both a privilege and a challenge for me. Speaking of privilege, creating a thesis in collaboration with Volkswagen Group and the Technical University of Dortmund allowed me to make the best out of both sides. I am grateful for the chances at Volkswagen Group to access actual industrial automated driving projects and corresponding resources, and for the convenient exchange with domain experts, which enabled the development of this thesis, considering actual industrial challenges and goals. I am also thankful for the remote support from the AQUA research team at the Institute of Informatics at the Technical University of Dortmund, which granted me maximum independence and flexibility for my research. Concerning challenges, staying away from a typical academic environment was not always easy. I want to thank many people who carried the challenges on with me along my overall PhD journey: Firstly, I sincerely appreciate the support from my colleagues at Volkswagen. I thank Dr. Fabian Schuldt for creating the initial research topic and my PhD candidate position at Volkswagen Group. I also want to thank Dr. Constanze Hungar for agreeing to be my industrial supervisor, who supported me in various research discussions and company processes. Besides, I want to thank my colleagues Dr. Lukas Hartjen, Yasin Bayzidi, and Andreas Bussler for the valuable exchanges and inputs that further shaped my thesis. I especially want to thank my peer, colleague, and good friend Dr. Robin Philipp, who continuously inspired me in topics, motivated me in difficult times, and cooperated with me in multiple publications and their corresponding crunch times. Moreover, I want to thank two former intern students, Yongqi Zhao and Weidong Hu, who diligently supported me in the experiments. Furthermore, I am deeply indebted to my university supervisor, Prof. Dr. Falk Howar, who paid great attention to all of my research work, contributed to my publications, and showed much empathy and understanding each time I struggled with difficulties. I am also extremely grateful to have Prof. Dr. Martin Törngren agree to be my second university supervisor. I truly enjoyed every chance to talk with them and value their opinions in research and life. At the end, I want to thank my parents for supporting me unconditionally and my dog Lamei for accompanying me working through many late nights.

Contents

List of Figures

List of Tables

Part I.

Introduction

1. Motivation and Research Goal

Over the past few decades, automated driving technologies have evolved steadily—from semi-automated, low-speed driving trials in the 70s and 80s (e.g., [41, 59, 111]), to the first long-distance, high-speed automated driving demonstrations in the 90s (e.g., [86, 89]), and now, to various companies conducting extensive public road tests (e.g., Waymo [62], Zoox [76]. Volkswagen [57]). Since the development of automated driving has made significant progress from prototypes to large-scale fleet testing, the research focus has gradually shifted towards the safety and reliability of the systems.

An intuitive opinion is that automated vehicles should perform at least as safely as experienced human drivers. Such an opinion is becoming increasingly acknowledged on an international, regulatory level. Exemplarily, the Automated Vehicles Act 2024 of the United Kingdom mandates that self-driving vehicles shall "achieve a level of safety equivalent to, or higher than, that of careful and competent human drivers" [112, 1.2 (2) a]. Similarly, the European Commission states in Implementing Regulation (EU) 2022/1426 that the manufacturer shall define the acceptance criteria from which the validation targets of the ADS are derived. Based on that, the residual risk for the operational design domain (ODD) shall be derived from accident data or data on performance of competently and carefully driven manual vehicles [27, Annex II].

However, verifying such a safety level of ADS is a nontrivial task. Unlike many other safety-critical systems (e.g., nuclear reactor control systems, rail signaling systems, airborne collision avoidance systems), ADS are envisioned to operate in very complex open environments, involving infinite possible situations. As Stellet et al. [102] also point out, gaps among expectations, specifications, and implementations cause fundamental challenges for the safety validation of ADS in an open environment. Exemplarily, society would expect that ADS shall not endanger vulnerable road users. As part of the ADS specification process, this expectation needs to be further broken down into technical requirements, like maintaining a safe distance of 1.5 m to cyclists (cf. [33]). In reality, it is still possible that either the 1.5 m safety distance is not correctly implemented (due to a so-called *implementation gap*) or the safety distance is sufficiently high. However, the cyclist still feels endangered during overtaking (e.g., by being passed with a too high velocity due to a *specification gap* despite sufficient lateral distance). Such gaps should be continuously identified and addressed within the ADS lifecycle.

Traditionally, the safety of vehicles with lower automation (e.g., with advanced driver-assistance systems) is validated by mileage-based testing [39]. However, ADS come with higher reliability requirements due to higher automation levels and the absence of a fallback human driver. A mileage-based validation for ADS is not feasible anymore due to two aspects:

High amount of required driving distance. Hundreds of millions of miles are required to show that the accident avoidance of ADS is as good as that of a human driver. Besides, drives to accumulate mileage would need to be repeated for software or hardware updates. [119, 58]

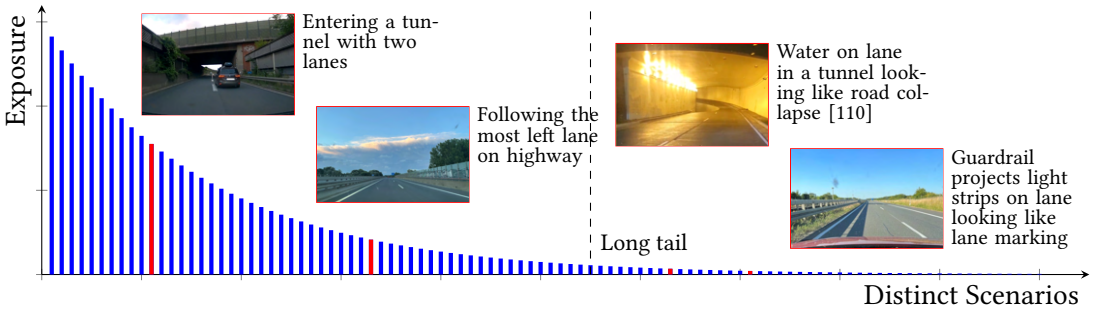


Figure 1.1.: Long tail distribution of distinct scenarios exposure

Randomness of encountered situations. While the static aspects of traffic (like road geometries, infrastructures) can still be planned as part of tests, the dynamic aspects (like interactions with other traffic participants on public roads) cannot be controlled. Furthermore, situations with a low occurrence could be missed. This is problematic for situations that could lead to hazardous behavior of ADS. In statistics, this phenomenon is called the long tail. Mileage-based testing of ADS is also subject to the long tail nature. Figure 1.1 illustrates the long tail phenomenon in a traffic scenario distribution.

To deal with these two aspects, scenario-based testing has been introduced in the last few years. So-called *scenario-based* means to define a finite set of relevant driving scenarios from dynamic traffic situations [46]. Instead of exclusively relying on accumulating mileage in real road tests, scenario-based tests can be conducted in simulation or on proving grounds and thus enable more testing within a limited time. Most importantly, scenarios for testing can be defined comprehensively (covering specific low-probability situations) and can be traced back for regression. Thus, scenario-based testing can potentially complement random validation drives by targeting the long tail. However, scenario-based testing comes with the dependency of specifying a relevant scenario catalog.

It is essential to analyze how ADS functions and components could be exposed to a hazard in their operational environment to comprehensively define relevant scenarios and especially critical ones. Three dimensions should be thoroughly understood and utilized, namely the operational environment, the functions and components of ADS, and the potential hazardous behaviors of ADS. More specifically, it needs to be considered what the environment consists of, how it can be structured, and which environmental conditions can be an external root cause for ADS not to operate safely. Meanwhile, it is important to analyze what components ADS consist of and in which ways these can be affected. Finally, the possible hazardous behaviors ADS can perform should be investigated. Besides collisions with road users, other hazardous or non-compliant behaviors should also be considered, which increase risk in traffic.

There exist many studies on the creation of critical scenarios, which cover these three dimensions to different extents: One group of studies (e.g., [118, 4, 120, 63]) generates critical scenarios by optimizing criteria like safety metrics (cf. time-to-collision, safety distances, driveable area). They mainly touch on the aspect of the potential hazardous behavior of ADS. However, the construction of these scenarios focuses on interaction with other traffic participants with a basic road geometry, while other components of the environment, as well as the discussion

of ADS functions are omitted. Another approach is considering human driver accident records and reconstructing the corresponding scenarios (e.g., [125, 127, 35]). They are advantageous to restore difficulties in the environments, but at the same time are biased towards human drivers. They only cover near collisions or collision situations and thus lack the possibility to reveal difficult conditions that trigger other potential hazardous behaviors. Until now, only human drivers and traditional vehicles have been involved in these accidents, so that ADS and analysis of their components are not of relevance to these works at all.

Instead of directly deriving critical scenarios based on the methods above, an alternative approach is to identify environmental conditions that could lead to hazards and integrate them into various scenario contexts. This approach can ultimately contribute to deriving potential critical scenarios in the long tail and addressing the aforementioned specification and implementation gaps (cf. Stellet [102]) by testing the ADS under these scenarios. It is also possible to achieve the desired comprehensiveness, when the three aspects of operational environment, ADS components, and hazardous behaviors are explicitly considered in the identification of difficult environmental conditions. Besides, this approach has the potential to accelerate the verification process: Constructing test cases based on difficult environmental conditions naturally assigns the tests with a focus on specific aspects in the scenario and specific driving functions, which improves the explainability of the test results and facilitates the system debugging process.

The identification of difficult environmental conditions is only part of the story. The other part is to utilize the identified conditions for verification and validation (V&V) activities. How to systematically generate test scenarios for difficult environmental conditions to argue that they are sufficiently tested is a problem that follows on the technical level. For example, an environmental condition *a halting vehicle with a hazard flasher in front* can be not only relevant in a lane-following scenario, but also interesting in the context of turning at junctions or entering a tunnel.

On the tactical level, testing of difficult environmental conditions should be integrated into the overall V&V strategy. In practice, the development and verification of automated vehicles rather corresponds to an iterative process [50] instead of simply following a V-model [91], as ADS have not yet achieved market maturity and have thus never been mass-produced til now. ADS and their components are developed, tested and updated in different versions with an increasing amount of features, continuously improving the system's overall performance. Though it is not often spoken of, we have observed a mismatch of required time for system development and for system verification in industrial ADS projects. Rapid developments of machine learning, sensor technologies and computational power can quickly lead to frequent changes during system development. Fixing a bug with a new software release can be especially quick. On the contrary, verifying the overall functionality of ADS after integrating a change in the system can be much more complex and time-consuming, since ADS consist of many driving functions and components. Considering the iterative ADS development process, the mismatch between the required time of development and verification can propagate, and thus the gap can grow even greater. Therefore, the V&V strategy should be dynamically adapted according to the newest system implementation. Correspondingly, this requires updating the relevant difficult environmental conditions for each development iteration. Exemplarily, a system de-

tecting traffic signs not by vision but based on a map does not need to be tested against adverse conditions for traffic sign visibility (e.g., stickers on signs [72]). However, if in a new release the system is enhanced by a vision-based sign detection, a sticker on traffic signs causing an incorrect sign recognition by the camera can confuse the ADS. It should thus be brought into focus of the tests. Nevertheless, how the identification and testing of difficult environmental conditions should be synchronized with the overall ADS development and V&V process is not yet well-reflected from the existing studies about scenario-based testing.

Following the direction of scenario-based testing, this thesis works on the aforementioned open questions around the concept of difficult environmental conditions as a new contribution to approaching the long tail problem in the verification process of ADS. Since this thesis is created within an industrial context, we also aim to show compliance of our methodology with state-of-the-art type-approval regulations like EU 2022/1426 and conformance to applicable standards for assuring the safety of the intended functionality of ADS like ISO 21448.

1.1. Research Scope

With the understanding of the previously mentioned challenges, the major objectives of the thesis can be depicted as:

- interpreting the nature of difficult environmental conditions and providing a formalization solution
- developing a systematical method to identify difficult environmental conditions for ADS
- introducing a structured approach to integrate difficult environmental conditions into testing activities in an iterative ADS development process

In order to define a feasible scope of the research, certain delimiters are set before working towards the defined research objective:

1. The thesis works on concepts and methods for verifying the safety of highly/fully automated vehicles (or driving systems above level 3 of automation according to SAE standard [1]). Namely, direct in-cabin human involvement or control is excluded. Therefore, the methods discussed in this thesis may not be completely suitable for analyzing and evaluating driving systems with lower level automation, e.g., Advanced Driver-Assistance Systems (ADAS).

2. Cyberattacks from the environment are categorized in the research field of security. Therefore, they are excluded from the scope of the thesis due to its focus on safety.

3. E/E failures, software/hardware malfunction and functional insufficiencies of ADS can all lead to vehicle-level hazardous behavior. While the first two cases are addressed in functional safety [51], the thesis focuses on the last case, which is concerned mainly with the safety of the intended functionality [50]. Accordingly, the discussion in this thesis builds on the assumption that functional safety [51] is addressed and thus malfunctioning or E/E failures are irrelevant to the analysis.

4. The thesis discusses the ODD, critical scenarios, and difficult environmental conditions related to the capabilities and insufficiencies of ADS. The results can be utilized as meta-knowledge (e.g., which scenarios could be risky for the system and which component could be weak under certain environmental conditions) for real-time monitoring and assessment of potential risks and generating a safe driving policy during the operation (cf. Precautionary Safety Policy discussed in [21, 42] and Predetermined/Dynamic Risk Assessment in [122]). Still, the primary scope of the thesis does not include the discussion of risk assessment and safe driving policy, but rather the collection of the knowledge to enable these.

5. Some concepts proposed in this thesis indirectly touch the research topic *Safety Assurance* and *Safety Case*. E.g., in the illustration of a testing process in Chapter 9. These proposed concepts can be further developed for deriving safety argumentation and evidence. However, they are not discussed in depth and are not the focus of this thesis.

6. The thesis illustrates how ADS functions can be included in the identification and formalization of difficult environmental conditions. When detailed system knowledge is available, including concrete ADS functions in the corresponding processes is beneficial. The discussed ADS functions in this thesis are kept on a general level, which can be mapped to more concrete, specific ADS in real practice. Thus, the analysis of concrete software/hardware/function or system design insufficiencies of specific ADS is also not in the scope of the research.

1.2. Research Questions

For tackling defined research objectives in the previous section, we derive five concrete research questions with corresponding rationales:

RQ 1: [How should the nature of difficult environmental conditions be understood?]
Environmental conditions are understood differently in literature, although they are often mentioned in the same context of safety verification of automated driving functions or systems. Besides, there is also no formal definition of what is ”difficult” in the environment for ADS. However, an understanding and formal definition of difficult environmental conditions is a fundamental step prior to their identification, management, and utilization.

RQ 2: [How should difficult environmental conditions be formalized and managed?]
The operational environment is an open world that consists of infinite conditions. Naturally, it can be assumed that the amount of difficult environmental conditions is also huge. Especially during the early development iterations, ADS could contain more flaws and insufficiencies, and even a trivial environmental condition can appear difficult to the system. Methods to manage the collections of difficult environmental conditions are necessary for simplifying the subsequent use of them (e.g., quick search based on a

given criterion, calculating a coverage index). Meanwhile, there can be diverse stakeholders of difficult environmental conditions. There could be multiple organizations/experts involved in the identification work. An aligned documentation of the identification results can facilitate cooperation and simplify the review. Also, users of the difficult environmental conditions (e.g., test case designers, regulatory inspectors) should be able to understand these artifacts for their purpose of use.

RQ 3: [How can difficult environmental conditions be systematically identified?]
With increasing autonomy levels of vehicles, the origins of risks are significantly shifting from inside the driving system to outside. Traditional safety verification and validation (V&V) methods focusing on internal system failures are insufficient for spotting the risks arising externally from the complex environment. Novel V&V concepts aim at comprehensively testing automated vehicles, especially in difficult environmental conditions, as a basis for structured safety arguments.

RQ 4: [How should difficult environmental conditions be converted to potential critical scenarios?]
Difficult environmental conditions are ultimately used for evaluating the performance of ADS and exposing the system insufficiencies. However, they cannot be tested standalone. They need to be converted to scenarios and corresponding test cases.

RQ 5: [How can the traceability of difficult environmental conditions be established in the development lifecycle of ADS according to state-of-the-art standards and regulations?]
The development and verification of ADS is an iterative process. Test results from the current iteration will impact the design of ADS in the next iteration. In this way, a difficult environmental condition will not be difficult for every system version. It is important to trace the testing results of each difficult environmental condition to determine the test focus of the subsequent iteration, so that the utilization of testing resources can be optimized. At the end, it is important to argue for sufficient test coverage of difficult environmental conditions.

1.3. Thesis Structure

This thesis is primarily based on previously published papers (including Paper I, II, III, and IV) and is further supported by important yet unpublished results obtained during the research. To ensure a coherent and logically unfolding narrative, the thesis is structured into five main parts. Part II through Part IV constitute the core research content, with each part addressing one or more research questions outlined in Section 1.2. Each part is divided into multiple chapters, and the corresponding papers are incorporated into these chapters accordingly. Table 1.1 provides an overview of the thesis structure, including the organization of parts, chapters, research questions, and associated papers. Brief descriptions of each part are presented as follows:

Part I: Introduction. This part introduces the motivation and goal of this thesis. Based on this, a research scope and five research questions are defined. Research methodology and major

Table 1.1.: Thesis organization

Part I	Introduction		
Part II	RQ 1	Chapter 3	Paper I
	RQ 2	Chapter 4	Paper I, III
Part III	RQ 3	Chapter 5	Paper I, III
		Chapter 6	Paper III
		Chapter 7	Paper II
Part IV	RQ 4	Chapter 8	Paper IV
	RQ 5	Chapter 9	Paper IV
Part V	Conclusion		

contributions are briefly summarized. Furthermore, related work is collected and discussed according to the research questions. The part unfolds into Chapter 1 and Chapter 2.

Part II: Nature and Formalization of Difficult Environmental Conditions. This part discusses the essence of difficult environmental conditions according to the state-of-the-art relevant terms, answering research question **RQ 1**. It further demonstrates a concept of the formalization of difficult environmental conditions, answering research question **RQ 2**. The part consists of Chapter 3 and Chapter 4.

Part III: Elicitation of Difficult Environmental Conditions and Related Scenarios. This part demonstrates a knowledge-driven and a data-driven method to elicit difficult environmental conditions or their corresponding scenarios, tackling research question **RQ 3**. Besides, findings based on the identified difficult environmental conditions are discussed. The part includes Chapter 5, Chapter 6, and Chapter 7.

Part IV: Testing and Evaluation of Difficult Environmental Conditions. This part elaborates a method to convert potential difficult environmental conditions into test scenarios, answering research question **RQ 4**. It further introduces the testing of difficult environmental conditions in the overall development and verification process of ADS. Concrete requirements are presented to establish traceability in the process, answering research question **RQ 5**. The part comprises Chapter 8 and Chapter 9.

Part V: Conclusion. This part discusses the major contributions and limitations of the thesis by reflecting on the findings of the proposed research questions. An overview of possible future work based on this research is provided. The part consists of Chapter 10 and Chapter 11.

1.4. Research Methodology

As displayed in Section 1.3, the main body of this thesis includes Part II, III, and IV, which unfolds into Chapter 3 to Chapter 9. Each Chapter serves one or multiple specific research objectives. Common research methods in software engineering were selected to fulfill each research objective. Table 1.2 provides an overview of the adopted research methods according to the chapters and their research objectives: Chapter 3 aims to propose and explain a new con-

cept related to existing concepts and terms. With an exploratory nature, literature review and grounded theory are chosen for this chapter. Chapter 4 to Chapter 9 target developing methods, tools, or processes for understanding the problems around the proposed concept. Thus, design science research is a major method for these chapters. Besides, the developed artifacts are evaluated with proper methods like interview, field study, and case study evaluation.

Table 1.2.: Research methods based on chapters

Chapter		Objective	Method
3	Terms and Definitions	Understanding the nature and the purpose of difficult environmental conditions	Literature review and grounded theory
4	Formalization of Difficult Environmental Conditions	Deriving machine-readable formulations for describing difficult environmental conditions	Design science, prototyping, and interview
5	Knowledge-Driven Approach to Identify Difficult Environmental Conditions	Developing a systematic designated method for identifying difficult environmental conditions	Design science
6	Practice of the SHFA Method and Findings	a. Evaluation of the SHFA method b. Collecting difficult environmental conditions c. Analyzing the distribution of the identifications	Interview and field study
7	Data-driven Reconstruction of Disengagement Scenarios	Developing an automatic method for generating critical scenarios with difficult environmental conditions based on real test drive data	Design science, prototyping, and case study evaluation
8	Injection of Difficult Environmental Conditions into Scenarios	Developing a systematic method for converting difficult environmental conditions into test scenarios	Design science and case study evaluation
9	Conformance, Compliance, and Traceability in the Test Implementation	Proposing concepts for maintaining traceability of difficult environmental conditions throughout the V&V process with conformance to ISO 21448 and compliance with EU 2022/1426	Design science and case study evaluation

1.5. Research Contributions

In general, this thesis offers concepts and methods for ADS safety validation, which can be mostly mapped to the activities in the top right corner of the well-known V-model [91] for system test and validation. As discussed throughout the next chapters, the research also contributes to Clause 7, 9, and 10 of the ISO 21448 [50]. Last but not least, Figure 1.2 maps the contributions by chapters to the principles to be followed to derive scenarios of the European implementing regulation (EU) 2022/1426 [27], which needs to be adhered to to receive a European type approval for automated vehicles.

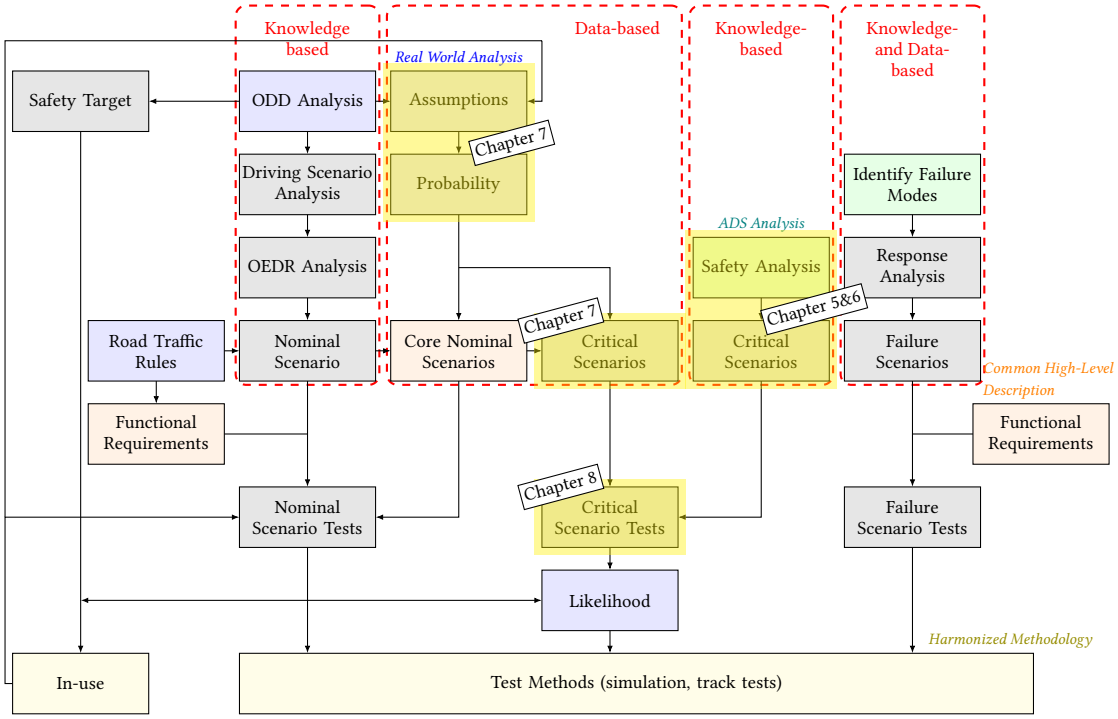


Figure 1.2.: Principles to be followed to derive scenarios relevant for the ODD of the ADS (figure taken from (EU) 2022/1426 [27]). Parts that this thesis contributes to are highlighted in yellow (knowledge and data-based identification of critical scenarios and corresponding tests).

On a more detailed level, according to each proposed research question, the major contributions (a) to (l) can be summarized as follows:

RQ 1: [How should the nature of difficult environmental conditions be understood?]
This research question leads to (a) an elaboration of the state-of-the-art terms and their relations in the context of safety verification of ADS, (b) a proposed formal definition of difficult environmental conditions.

RQ 2: [How should difficult environmental conditions be formalized and managed?]
This research question leads to (c) summarized three major types of difficult environmental conditions as the presence/absence, the behavior, and the interaction of environment factors, (d) a developed formalization according to each type, and (e) analyzed further possibilities to support the management of difficult environmental conditions, e.g., considering descriptive vocabularies based on ODD taxonomies.

RQ 3: [How can difficult environmental conditions be systematically identified?]
This research question elicits (f) a knowledge-driven, analytical method called Scenario-based Hazard and Fault Analysis (SHFA) to identify difficult environmental conditions, (g) an industrial practice of the SHFA methods, producing a catalog of identified difficult

environmental conditions, (h) a data-driven, automatic pipeline to reconstruct critical scenarios based on automated vehicle disengagements.

RQ 4: [How should difficult environmental conditions be converted to potential critical scenarios?]
This research question guides towards (i) concepts to derive critical scenarios based on difficult environmental conditions and nominal scenarios (cf. [27]), (j) a systematic method to implement the concept and accordingly a simulation-based case study.

RQ 5: [How can the traceability of difficult environmental conditions be established in the development lifecycle of ADS with the conformance to state-of-the-art standards and compliance to regulations?]
This research question leads to (k) an analysis and interpretation of test requirements in a realistic ADS V&V process with conformance to state-of-the-art standards (like ISO 21448), and (l) proposed concepts to establish traceability of difficult environmental conditions in the iterative development and verification of ADS.

2. Related Work

This chapter provides a roadmap of standards and regulations involving in the thesis and discusses the state-of-the-art contributions according to each proposed research question in Chapter 1, namely understanding and formalization of difficult environmental conditions, elicitation of difficult environmental conditions, elicitation of difficult environmental condition related scenarios, method and process for integrating difficult environmental conditions in the testing, and traceability in the testing of difficult environmental conditions.

2.1. Roadmap of Standards and Regulations

Since this thesis has been created in Germany, the following sections touch on the most relevant legislation and standards for introducing ADS to German and European markets. While we are aware of the fact that there are further efforts towards legislation and standardization in other countries and markets, an exhaustive enumeration of these is beyond the scope of this thesis.

2.1.1. Legislation on ADS

Commission Implementing Regulation (EU) 2022/1426 of August 2022 [27] lays down rules for the type approval of the automated driving system of fully automated vehicles at the European level. The regulation provides general, technical, and performance requirements and tests for ADS and expects the manufacturer to supplement these with sufficient documentation demonstrating the absence of unreasonable risk. Moreover, principles for deriving scenarios relevant to the operational design domain (ODD) are outlined. In the context of introducing ADS to the German market, an EU type approval for fully automated vehicles in small series production can be issued by the German Federal Motor Transport Authority (”Kraftfahrt-Bundesamt (KBA)”) following the rules of the (EU) 2022/1426 [37].

Another possibility to receive a German national type approval is to follow the German regulation AFGBV [32], which is the abbreviation for Autonomous Vehicles Approval and Operation Ordinance (”Autonome-Fahrzeuge-Genehmigungs-und Betriebs-Verordnung”). The regulation came into force in July 2022 and implements the sections of the German Road Traffic Act (”Straßenverkehrsgesetz (StVG)”) regarding series approval of ADS. The AFGBV governs operation, approval and testing of fully automated vehicles inside their defined operation areas. It also provides general, technical, and performance requirements and defines methods for testing and validation of ADS. The German Federal Motor Transport Authority can issue German national type approvals for fully automated vehicles by following the rules of the AFGBV [38].

The German Road Traffic Ordinance (”Straßenverkehrsordnung (StVO)”) [33] regulates traffic in Germany by providing a set of traffic rules, which need to be adhered to by any participant

in public traffic, including automated vehicles. The (EU) 2022/1426 [27, Annex II, 1.3] explicitly requires ADS to comply with the traffic rules of the country of operation. The United Kingdom's Automated Vehicle Act [112] from 2024 provides a legal framework for automated vehicles within the United Kingdom. It addresses regulatory schemes for automated vehicles and liability aspects like responsibility of the vehicle manufacturer, developer, and operator. The act also provides power to update type approval requirements. However, as of mid-2025, there are no regulations in force yet laying down rules for the type approval of automated vehicles in the United Kingdom. The required amendments to existing type approval regulations are announced to be in place at the end of 2026 [23].

The United Nations Economic Commission for Europe (UNECE) is responsible for the development of regulations and norms to facilitate economic integration. Specifically regarding ADS, the *Working Party on Automated/Autonomous and Connected Vehicles (GRVA)* is pursuing regulatory work [115]. One of the regulations developed in this context is the UN Regulation No. 157 [116], which governs how automated lane keeping assistance systems (ALKS) should perform (e.g., specific safety distances). The regulation was originally published in 2021 and has been updated several times since then. The UN R 157 is also referred to by the (EU) 2022/1426 regarding lane change and pedestrian crossing scenarios and parameters.

2.1.2. Standards on ADS

The standard SAE J3016 [1] titled "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles" provides a classification of six different automation levels for automated driving systems and was first released in 2014. The classification consists of the levels "no driving automation", "driver assistance", "partial driving automation", "conditional driving automation", "high driving automation", and "full driving automation". Whereas the first three levels of systems assist the driver and require to be constantly monitored, the latter three are responsible for monitoring the environment and performing the driving task accordingly.

The ISO 21448 [50], titled "Road vehicles — Safety of the intended functionality (SOTIF)", was released in 2022 and provides guidance on the identification and mitigation of unreasonable risk, which can originate from functional insufficiencies or reasonably foreseeable misuse of ADS. Risks resulting from electrical and/or electronic failures or vehicle security vulnerabilities are not in scope of the ISO 21448. The standard provides an iterative process consisting of several analytical and verification & validation (V&V) activities. Due to the industry's need for methods to ensure the safety of ADS and the novelty of the ISO 21448, SOTIF has become a central topic of research and standardization efforts in recent years. This thesis, with all of its concepts, methods and case studies can also be been as such an effort and thus another step towards safe ADS.

The ISO 26262 [51], titled "Road vehicles — Functional safety", was released in 2011 and revised in 2018 and deals with functional safety of electrical and/or electronic systems integrated into road vehicles. One of its key aspects is the introduction of the automotive safety integrity level (ASIL) as a risk classification scheme. Based on a hazard and risk analysis (HARA), ASILs are determined, which result in different safety requirements, e.g. regarding the reliability of hardware. With the increasing automation and thus growing scope of driving systems within

the last decade, it became obvious that functional safety alone is not sufficient anymore to assure safe systems, resulting in the ISO 21448 being published to address the safety of the intended functionality.

The ISO 3450x series consists of five standards on test scenarios for automated driving systems. ISO 34501 (2022) serves as a dictionary to align terms and definitions, and ISO 34502 [52] (2022) deals with a scenario-based safety evaluation framework. ISO 34503 [53] (2023) provides a taxonomy and structure for operational design domains, ISO 34504 [54] (2024) offers a classification of scenario parameters. ISO 34505 is currently under development with the final draft (FDIS) being in the approval phase as of mid-2025. The ISO 34505 is planned to define a methodology for scenario evaluation and test case generation.

The ISO/IEC/IEEE 29119 [55], titled "Software and systems engineering — Software testing", was released in 2013 and revised in 2022, and provides standards for software testing. These standards cover test process descriptions, test documentation and test design techniques.

The UL 4600 [113] "Standard for Safety for the Evaluation of Autonomous Products" lays out topics that shall be included within a safety case for ADS. Topics range from engineering processes and risk assessments over V&V and tool qualification to lifecycle concerns and maintenance. The extensive standard is intended to provide a framework in which other standards, like the ISO 26262 or ISO 21448, can fit into.

2.2. Understanding and Formalization of Difficult Environmental Conditions

Understanding the operational environment of ADS is a preliminary yet crucial step for designing ADS and the corresponding safety verification. In the research field of autonomous driving, the terms *environment* and *environmental conditions* are often interpreted differently in the literature. As observed within different papers [92, 94, 43, 30, 53], a common interpretation observed tends to limit environment to the natural world and its phenomena like illumination and weather. In this thesis, the environment is discussed in a broader scope. That is, the natural environment and everything surrounding the system boundary (including other objects, e.g., traffic participants, and their interactions) are counted as a part of the environment.

In the development and verification process of ADS, the environment is often represented by the concept *scenario*. Therefore, research works defining and decomposing the scenario instead of the environment are considered in the development of this thesis: Ulbrich et al. [114] discuss the relation and differences among scene, scenario, and situation. They define scenarios as temporal progression between multiple scenes in a sequence and levels. To systematically determine the content of scenarios for ADS, Schuldt et al. [98] propose a general model that divides scenarios into the basic road network, situation-specific adaptations of the basic road network, descriptions and controls of actors, and environmental conditions. The model was later implemented in a German federal project called Pegasus [36] and is extended by Sauerbier et al.[96] with two more layers, namely, traffic infrastructure and digital information. Scholtes et al. [97] provide detailed descriptions of the latest six-layer model. The six-layer scenario description model is adopted in the thesis for systematically describing the environment for ADS, which is used as a foundation to elicit difficult environmental conditions later.

Besides, there exist diverse works about the formalization and standardization of scenarios to facilitate their design, implementation, and utilization. Firstly, Bock et al. [13] introduce a method called stiEF with an embedded textual domain-specific language (DSL) for the iterative evolution of textual scenario descriptions. They implement the DSL considering predefined domain-related vocabularies (e.g., *vehicle*, *main road*, *drive*) and relevant data structures. Their method supports coverĺng potential content of scenarios with five layers (cf. [98]) on diverse abstraction levels (cf. functional, logical, and concrete levels, cf. [74]). In this way, users of stiEF can efficiently describe scenarios according to the desired abstraction level based on the requirements of a specific ADS development phase, while the embedded DSL avoids potential vagueness and duplications in the descriptions. Similarly, Fremont et al. [34] propose a probabilistic programming language Scenic for the design and analysis of machine-learning-based, cyber-physical systems. They simplify the language syntax for writing complex scenarios and incorporate sampling techniques directly into the language. Users can follow the syntax to program scenarios with either hard or soft constraints, such as explicitly defining the range of a target vehicle's position or just specifying the existence of a target vehicle in a specific lane. Concrete scenarios can then be generated by sampling from a probability distribution under these predefined constraints. Furthermore, due to the rapid development and growing application of simulation techniques in ADS testing, there is a strong demand for standardized scenario descriptions for simulation purposes. Standardization is essential to enable the exchange and usability of scenarios across different simulation platforms. Two well-known, publicly developed standards in the industry are the OpenSCENARIO standard [8] and the OpenDRIVE standard [7]. The OpenSCENARIO standard provides specifications and file schemas for describing the dynamic aspects of a scenario, such as maneuvers and the involved entities, including vehicles, pedestrians, and other traffic participants. Meanwhile, the OpenDRIVE standard specifies the format for describing static road networks in scenarios, detailing elements such as roads, lanes, junctions, and static objects on the road. The aforementioned standardization and formalization efforts are all highly relevant to this thesis. They are either directly used in the method development and experiments (cf. Chapter 5, Chapter 7, and Chapter 8) or provide inspiration for the formalization of difficult environmental conditions (cf. Chapter 4).

2.3. Elicitation of Difficult Environmental Conditions

As ADS are an integration of multiple automated driving functions and components, many studies have investigated environmental influences on the performance of individual components on perception, planning, and action levels.

On the perception level, Ren et al. [92] investigate the effect of environmental influences like darkness and blurred and rotated camera images on object detection. Peng et al. [81] evaluate the uncertainty of object detection algorithms and identify different environmental influences (e.g., fog, snow, and unusual object appearances). Linnhoff et al. [71] provide an extensive catalog of sensor effects and demonstrate how cause-effect chains can be modeled, covering aspects such as emission, signal propagation, reception, pre-processing, and detection/feature/object identification. An effect like *false negatives in the object list* can, under unfavorable circumstances, lead to hazardous behavior, whose cause can be identified as a difficult environmental

condition. Their specified causes for the sensor effects are kept general (e.g., *pose of object parts*). Breitenstein et al. [15] systemize corner cases for perception into different detection complexity categories and provide corresponding examples. Concrete examples from the *Domain Level* to the *Scenario Level* can also be understood as difficult environmental conditions, like *a fence on the street* or *a person suddenly walking onto the street.* Tong et al. [108] design on-field tests to examine performance limitations of visual sensors under specific light conditions, where they validate four difficult environmental conditions for their recognition algorithm, namely low illuminance, lateral movement of the object, large longitudinal distance of the object, and high beams of the oncoming vehicle. In addition, Huang et al. [47] deduce difficult environmental conditions from system performance limitations. In their case study, they analyze the obstacle avoidance function of their system under analysis and identify thin objects (e.g., poles, tree branches) as potential difficult environmental conditions due to the limited angular resolution of their system's LiDAR sensors. Xing et al. [124] propose ontologies for identifying perception-related triggering conditions[1] based on a propagation chain of event models. Correspondingly, one ontology for triggering source is constructed based on the interactive relationship between the traffic environment elements and the ego vehicle. The other ontology for describing different processing stages of the perception system is built according to general perception principles. Then, triggering sources are associated with certain perception stages to derive their triggering effects. Difficult environmental conditions are identified from the triggering source under analysis by validating the triggering effects. Adee et al. [2] utilize a Bayesian network to present the causal relation between scenario factors and performance limitations of perception systems. Relevant scenes are learned from a driving dataset so that experts can further identify possible difficult environmental conditions from the scenes. This work exhibits the potential to support the identification of difficult environmental conditions by efficiently selecting relevant scenes with statistical methods.

On the planning level, Althoff et al. [3] offer an extensive set of benchmarks for motion planning components. This set consists of dangerous scenarios (e.g., illegal cut-ins) recorded from real traffic or hand-crafted. The specific constellations of the surrounding traffic participants on the given road network can be understood as potential difficult environmental conditions for planning components. Sharma et al. [101] summarize the disadvantages of commonly used path, trajectory, and behavior planning techniques and models, based on which general difficult environmental conditions can be inferred, given the information of adopted planning techniques or models of an automated driving system.

On the action level, Stolte et al. [104] perform System-Theoretic Process Analysis (STPA) [68] to systematically examine malfunctions of actuators for deriving safety goals and functional safety requirements. In their experiment, they develop a reference control loop and corresponding control components, control actions, and safety goals. Then, unsafe control actions and their causal factors are identified. For example, for the malfunction of a component *motion estimation via sensor*, they identify possible causes as *inadequate or missing feedback*, *feedback delays*, *measurement inaccuracies*, and *inadequate operation.* Although these causes are on component level, they can be a starting point for analyzing root causes from the environment.

[1] A concept defined in ISO 21884 [50], which is comparable to the concept of difficult environmental conditions in this thesis (cf. Chapter 3)

Besides the component level analysis, the identification of hazardous scenarios also contributes to identifying potential difficult environmental conditions. Kramer et al. [67] extend the Fault Tree Analysis (FTA) method to identify and quantify hazardous scenarios. During their process, potentially hazardous environmental conditions are exposed. By means of a global causal-effect analysis with establishing an environment model, a further holistic identification and expression of difficult environmental conditions is possible. Khastgir et al. [61] extend the STPA method to generate loss scenarios for testing purposes. Along the assumption process, some difficult environmental conditions can be indirectly identified. Based on an assumption of an unsafe control action of a specific system model, they enumerate the causes of it and decompose the causes into a wrong belief of a process model (*B1*), possible reasons for the model to have such a belief (*B2*), and further possible causal factors for *B2* (*B3*) to derive loss scenarios for testing. Coppola et al. [17] employ stochastic simulation and uncertainty modeling to quantify the impact of pre-selected situational variables on the pass-fail rate of ADS. The most influential situational variables for a failed test and their corresponding difficult environmental conditions can be identified.

There also exist a few contributions, which aim at structured techniques for eliciting difficult environmental conditions on the system level. Martin et al. [73] propose a workflow to identify difficult environmental conditions of relevant sensor technologies for an Autonomous Emergency Braking system. Zhao et al. [88] propose a method to iterate over all identified functional insufficiencies of an Adaptive Cruise Control (ACC) system and check whether any of the relevant elements of a given scenario could trigger an identified insufficiency. Their exemplary analysis identifies difficult environmental conditions such as *light intensity changes* for a camera or *crosswind* for actuators controlling the vehicle.

It can be seen from the aforementioned studies that major identification methods are on the component level and their focus lies on the perception components of ADS. We conjecture that the identification of perception-related difficult environmental conditions is relatively intuitive, as perception is the direct interface between the ADS and the scenario. Meanwhile, these studies are addressing various driving systems with different automation levels (ADAS, ACC, and ADS). Although they are valuable references, a direct fusion of their identifications will not be applicable for a specific ADS. A comprehensive and designated method to identify difficult environmental conditions for the overall sense-plan-act chain of a specific ADS has not been established as of the beginning of this dissertation. However, such a method is of high importance for analyzing weaknesses and insufficiencies of integrated ADS. Therefore, in Chapter 5, we propose a designated, comprehensive knowledge-driven method for this purpose and compare the method specifically to the works by Kramer et al.[67] and the STPA method [68] due to their similarities in the concept.

2.4. Elicitation of Difficult Environmental Conditions Related Scenarios

Being aware of the shortcomings of knowledge-driven approaches (e.g., human bias, unrealistic identifications), a data-driven approach is deployed in this thesis as a complementary consideration for the elicitation of realistic and difficult environmental conditions. Instead of directly

identifying the conditions, the goal is to obtain critical scenarios, which very likely contain such conditions.

To our best knowledge, data-driven critical scenario generation can be conducted in two ways in general: exploring the parameter space of roughly predefined scenarios (e.g., ego vehicle performs lane changing) using stochastic methods and reconstructing scenarios from real driving data. Previous works in the first direction show the potential of reducing the effort of scenario generation with the help of machine learning methods [10] or sampling techniques [129]. However, as a precondition, qualitative scenarios (cf. *scenario categories* [22]) have to be defined as fundament of the parameterization process, which is also a topic of current standardization efforts in ISO 34504 [54]. The other direction appears to be more straightforward, since scenarios are reconstructed by directly replaying or modifying automated driving data. The research work by Park et al. [80] presents a methodology to create challenging scenarios based on GPS data, on-board radar detections, and an HD map for Lane Keeping Assist System (LKAS) and ACC functions. Montanari et al. [77] attempt an automatic extraction of sequential, roughly parametrized maneuvers from bus data of a test vehicle and an automatic generation of scenarios by cascading the extracted maneuvers. This enables a further configuration or modification of maneuver details like duration and starting conditions. However, due to the high abstraction level of maneuvers and omitting kinematic details, the constructed scenarios may not be suitable for regression testing. Karunakaran et al. [60] illustrate a process of extracting road networks from LiDAR point cloud measurements and then, in conjunction with object detections, to extract lane change scenarios. They only consider the ego vehicle and one adversary vehicle related to the lane change maneuver to simplify the process. As a complementary work, Zofka et al. [135] additionally modify the temporal or spatial parameters of the lane-changing vehicle in the derived original lane change scenarios to increase criticality.

Regarding data selection, it is common to consider traffic accident or near-miss records (e.g., in the works [26, 125, 11]) for deriving critical scenarios, where the main participants are mostly only human drivers. This kind of data is advantageous in exposing the scenarios with relatively high operational complexity (e.g., encountering non-compliant traffic participants), which are constructive for testing automated vehicles at a similar difficulty level. However, human drivers' driving maturity and technology are not comparable to the current state of many automated driving systems. As a result, the traffic scenarios that are trivial for human drivers but challenging for automated driving systems could be neglected (or the other way round) if the focus is only put on the accident data. In fact, disengagement data where the main actor is the automated vehicle could directly reflect the problematic scenarios for the automated driving systems. Meanwhile, during the rapid prototyping of ADS, the disengagements occurring in the test drives are of interest for debugging and assessing driving functions. Also, the corresponding scenarios can directly be converted into test cases for regression testing after the function modification. For instance, Waymo presents how they reconstruct disengagement scenarios to conduct counterfactual simulations within their safety report of 2020 [99]. Based on these scenarios, they investigate the system performance assuming the human driver would not have intervened. Besides, other works [31, 121, 14] indicate that disengagement scenarios are highly relevant for investigating insufficiencies in the design of automated driving functions and the environmental root causes for hazardous behavior of ADS.

Inspired by these related works, we follow the straightforward approach (namely, reconstructing concrete critical scenarios from driving data) and introduce an automatic scenario generation pipeline in Chapter 7. To ensure that the reconstructed scenarios are relevant for ADS, ADS disengagement data are utilized as input.

2.5. Potential Critical Scenario Generation Based on Difficult Environmental Conditions

Given a catalog of identified difficult environmental conditions, there are two general approaches for integrating them into potential critical scenarios, namely *extension-based* and *combination-based*.

An extension-based approach means that new scenarios are crafted based on a specific testing focus by adding other scenario context from the ODD. The extension-based approach is intuitive and is followed by multiple previous works: Huang et al. [47] propose a framework to generate test cases based on difficult environmental conditions. Identified difficult environmental conditions are extended with certain operational situations to obtain the test scenario. As a follow-up work, Xing et al. [124] apply the framework and demonstrate 20 test cases with corresponding testing results. Rau et al. [90] illustrate how to use triggering events[2] to impose restrictions on the selection of scenario variables and derive concrete and SOTIF[3]-relevant scenarios based on general operating situations.

A combination-based approach means that scenarios are derived by combining the to be tested aspect with existing scenarios (so-called *reference scenarios* in the following). Mancher et al. [28] follow this concept for scenario-based SOTIF analysis. Firstly, they define reference scenarios within an ODD and ensure them to be hazard-free via testing. Then, potential difficult environmental conditions are combined with the reference scenarios to derive SOTIF-relevant scenarios. With an investigation into standards and regulations addressing scenario-based testing, we also observe that combination-based approaches are used to define critical scenarios. The U.S. Department of Transportation proposes a framework of test scenarios for ADS [107]. The framework specifies that common test scenarios shall cover four core aspects: tactical maneuver behaviors, ODD elements, *Object and Event Detection and Response* (OEDR) capabilities, and failure mode behaviors. Certain conditions of each aspect can be modified to increase the complexity of the test scenarios. The standard ISO 34502 requires identifying the relevant scenario space based on an ODD and determining critical scenarios with the help of risk factors [52]. Risk factors are understood as factors of a scenario whose presence increases the probability of the occurrence and/or the severity of harm. The first type approval document for fully automated vehicles Implementing Regulation (EU) 2022/1426 [27] specifies that nominal, critical, and failure scenarios shall be derived for testing [27]. Nominal scenarios are reasonably foreseeable situations based on ODD analysis, where the interactions of the ADS with other traffic participants are non-critical. Critical scenarios are partially derived considering edge-case assumptions on nominal scenarios. [27] The aforementioned *certain conditions*,

[2] *Triggering event* is the predecessor concept of *Triggering condition* from ISO 21448 PAS [56]
[3] Safety of the intended functionality [50]

risk factors, or *edge-case assumptions* can be understood as potential difficult environmental conditions. They are introduced to pre-defined nominal scenarios to add further risks and criticality for testing.

After comparing both approaches, we follow the combination-based approach and propose a method in Chapter 8, and define rules to conduct the combination process. To structure the potential critical scenario generation, we allocate the combination and parameterization process to three abstraction levels proposed by Menzel et al. [74], which are originally introduced for modeling scenarios for the development, testing, and validation of ADS.

2.6. Traceability and Conformance in ADS Development Cycles

In the context of testing of ADS, previous studies mainly provide solutions for (1) generating test cases, and (2) automating and formalizing test execution. Stepien et al. [103] introduce a methodology to generate test suites accounting for both severity and exposure. They extract critical scenarios and nominal scenarios from naturalistic driving data and abstract their parameter value distributions by distribution functions and regression models. Afterwards, they apply a heuristic search-based approach to generate test cases based on both types of scenarios by optimizing the objective functions for severity and exposure. They demonstrate the effectiveness of the proposed methodology with prototypical cut-in scenarios. Li et al. [70] illustrate two algorithms for automatically converting an ontology to a combinatorial test input model and generating test suites. They refer to a simplified road section ontology and demonstrate the applicability of the ontology-based test suite generation in the industrial context based on an autonomous emergency braking system function. Rocklage et al. [93] explain infeasibility in providing ADS under test with the same inputs in each test run in regression testing. As an alternative solution, they introduce concepts of *static scenario* and *hybrid scenario* as the foundation to generate test cases for the regression test to approximate the equality in the input space. They develop a backtracking algorithm, working with a motion planner to specify parameters of the corresponding test cases.

Meyer et al. [75] elaborate on the importance of the formalization of test cases and introduce a tool-agnostic XML-based test case specification format, which enables the specification of test attributes, test variables, pre-conditions, inputs, or test procedure, post-conditions, and expected results for scenario-based test cases. In the development process of the format, they consider requirements including versatility, test standard conformance, scenario integration, step- and event-based test procedure, machine-readability, modularity, and interoperability. Zhou et al. [130] propose a test framework, named AVUnit, to support automatic and systematic testing of ADS. The framework comprises two DSLs for formalizing test scenarios and specifications and test oracles to be monitored during the test execution, respectively. Besides, two algorithms are introduced to automatically search for test cases, which can potentially trigger violations of the specification and test oracles. The framework is demonstrated with an open-source ADS software stack, Apollo [9], and enables the identification of 19 insufficiencies within the Apollo stack based on generated failure test cases.

Nevertheless, the existing works mainly focus on the test phase and thus rarely consider the complete development life cycle of ADS. How the test cases are traced to requirements,

and how test results can impact the debugging and modification of ADS functions and further derive coverage indices are not sufficiently discussed. As the ISO 21448 standard displays, the development and verification of ADS is an iterative process (cf. [50, Clause 4, Figure 10]). To utilize difficult environmental conditions for the safety verification of ADS in conformance with state-of-the-art industrial standards like ISO 21448, their identification and testing activities should be allocated and updated in the iterative process. In Chapter 9, we elaborate on what the allocation looks like to fulfill requirements from ISO 21448 and propose concrete requirements to establish traceability.

Part II.

Nature and Formalization of Difficult Environmental Conditions

3. Terms and Definitions[1]

Establishing a clear understanding of difficult environmental conditions and their role in scenario-based testing is a fundamental prerequisite for their effective identification and utilization. This chapter lays the conceptual foundation by examining key terms relevant to safety verification and scenario-based testing in the context of automated driving. Through this exploration, the nature and purpose of difficult environmental conditions are clarified, culminating in a formal definition to guide the subsequent research.

3.1. Environment and Scenarios

ADS are bounded systems. Everything lying outside the boundary belongs to the *environment* [25] of the ADS and can be called an *environmental condition*. An environment is always relative to a system and is based on a subjective perspective of the system.

In the field of automated driving research, a similar term *scenarios* is widely recognized and used in both development and verification contexts. [74]. We refer to the commonly cited definition by Ulbrich et al. [114], which states that a scenario describes the temporal development between multiple scenes in a sequence. A scene, on the other hand, represents a snapshot of the environment, encompassing both the scenery and dynamic elements, along with the self-representations of all actors and observers (e.g., their skills, abilities, states, and attributes), and the relationships among these entities.

During the design and development phase, scenarios can be defined as part of the system specification. For instance, Automated Driving Systems (ADS) must be capable of performing an unprotected left turn at a junction, even in the presence of other road users. Scenarios are then designed to reflect various types of junctions, differing numbers of road users, as well as their positions and velocities. In terms of the verification and validation phase, scenarios can constitute test cases for ADS. The system must demonstrate its ability to complete tasks in concrete test scenarios, such as driving from Point A to Point B under specific constraints (e.g., within a limited time). Therefore, scenarios not only comprise certain environmental conditions but also include the intended actions of ADS and are described from an omniscient perspective. Scenarios used to develop and verify ADS essentially describe a set of environmental conditions and the corresponding maneuvers ADS are expected to perform. Although possible environmental conditions in an open world are infinite, scenarios must be limited in number and selected to cover the most relevant environmental conditions.

[1] This chapter is based on Paper I [132] and therefore contains verbatim content previously published (©2022 IEEE).

3.2. Scenario Taxonomy and Abstraction Levels

To minimize the number of scenarios while covering the most relevant environmental conditions, numerous studies have focused on identifying critical scenarios for automated driving. The definition of *critical scenarios* varies across these studies, as they often have different verification focuses and operational domain: To verify the safety on a component level, Klischat et al. [63] present an approach to generate critical scenarios for motion planning based on simulation data, where criticality is determined by the size and number of drivable areas for the ego vehicle. Ponn et al. [87] focus on highway scenarios and measure criticality using the Time-To-Collision (TTC) metric. Besides, they point out the inconsistent use of terms *challenging scenarios*, *critical scenarios* and *complex scenarios* in the literature. They consider challenging or complex scenarios to be those that are difficult for automated vehicles to safely operate, while critical scenarios are defined only through the occurrence of a critical situation during testing. Some works research scenario criticality on a microscopic level. For instance, Thal et al. [106] consider an unprotected left-turn scenario and use Time-to-Brake (TTB) as a metric for indicating criticality and generating corresponding safety-critical test cases.

With the goal of validation of the safety of automated vehicles as a whole (namely, neither focusing on verification of a specific component, nor limiting to specific domain or road geometry), the EU type approval regulation (EU 2022/1426) introduces a scenario taxonomy, requiring automated vehicles to be evaluated accordingly. In this context, the term *critical scenarios* is formally defined alongside two other categories: *nominal scenarios* and *failure scenarios*[2].

Nominal scenarios are defined as reasonably foreseeable situations encountered by the ADS when operating within its ODD, involving non-critical interactions with other traffic participants and ensuring the normal operation of the ADS. In nominal scenarios, environmental conditions should be favorable for a safe operation. Specifically, the behavior of other traffic participants should comply with traffic rules, and traffic conditions should be clear (e.g., not overly complex, with ideal weather conditions).

Critical scenarios, on the other hand, refer to edge cases (i.e., situations with an exceptionally low probability of occurrence) and operational insufficiencies. According to the context in EU 2022/1426, we believe that the so-called *operational insufficiencies* are an equivalent concept to *functional insufficiencies* in the standard ISO 21448. Specific environmental conditions that expose operational, i.e., functional insufficiencies, are key components of critical scenarios.

The scenario taxonomy above specifies the requirements regarding the content of a comprehensive scenario catalog. Subsequently, different scenarios (e.g., nominal and critical) should be systematically developed according to different process stages. In this regard, Menzel et al. [74] propose a top-down concept of developing scenarios along functional, logical, and concrete levels, which has continuously gained more recognition and acceptance in the community. They suggest that domain experts should first design abstract, functional scenarios in accordance with system specification, legal requirements, and safety concepts (e.g., functional safety). Then, scenarios should be parameterized according to quantifiable technical require-

[2]Failure scenarios mean the scenarios related to ADS and/or vehicle components failure, which may lead to normal or emergency operation of the ADS, depending on whether or not the minimum safety level is preserved. [27] They are outside the scope of this thesis, as they pertain to risk mitigation following unintended automated vehicle incidents.

ments from system development. Finally, scenarios should be instantiated with concrete values to execute tests for system verification and validation.

3.3. Functional Insufficiency

As discussed in the previous section, analyzing functional insufficiencies is a significant step in the identification of critical scenarios according to EU 2022/1426. ISO 21448 defines functional insufficiency as *insufficiency of specification or performance insufficiency*. Insufficiency of specification refers to an incomplete specification, such as an uncommon road sign that is not included in the system specification, which may cause the system to fail to detect it. Performance insufficiency pertains to limitations in the system's technical capabilities. Functional insufficiencies can manifest at various levels (system, function, and component) and may involve different components (perception, planning, and action). When the system is a white box, functional insufficiencies become observable as output insufficiencies at the interfaces between functions and components. One or more functional insufficiencies can result in hazardous behavior at the vehicle level once triggered. A case study of a Lane Keeping Assistant System (LKAS), provided in Table 3.1, illustrates different manifestations of functional insufficiencies. Functional insufficiencies can exist after the design and implementation of the system. How-

Table 3.1.: Insufficiencies on different abstraction layers (©2022 IEEE)

Level	Type	Example for LKAS
Vehicle	Hazardous behavior	Vehicle drives out of the corridor
System	Specification insufficiency	Detection of colorful lane marks is not considered in the system design
Function Interface	Output insufficiency	The detected lane mark has an offset from its real position
Function	Functional insufficiency	The detection of lane marks is not sufficiently robust regarding wet road surfaces
Component Interface	Output insufficiency	The generated region of interest is much bigger than usual
Component	Functional insufficiency	The camera is unable to distinguish between water reflection and real lane mark

ever, they may be either known or unknown at any given point in the system lifecycle [50]. For example, it is well-known that all sensor modalities have limited effective ranges, while it remains unknown whether an object detection algorithm can always detect target objects without further investigation. Known functional insufficiencies are addressed during the system specification and design phase. For instance, knowing that camera performance is limited in low light, LiDAR sensors can be added to the perception component to compensate. On the other hand, unknown functional insufficiencies must first be identified so that countermeasures can be developed. Achieving Safety of the Intended Functionality (SOTIF) can be seen as

a process of continuously uncovering and resolving unknown functional insufficiencies until the remaining risk is deemed acceptable.

3.4. Triggering Condition

Identifying unknown functional insufficiencies depends on testing the ADS under specific conditions. In this regard, ISO 21448 defines a ”pair concept” related to functional insufficiencies, called the triggering condition. As specified by the standard, functional insufficiencies are activated by triggering conditions, which impact the vehicle's behavior.

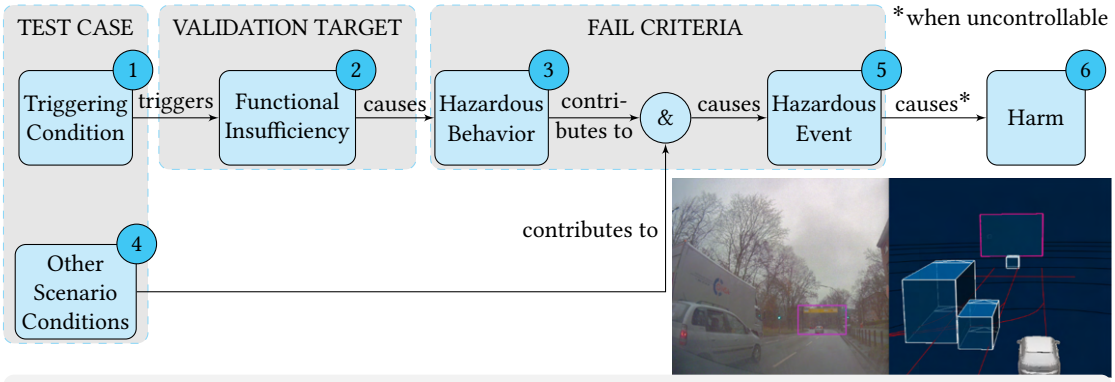
According to the definition, a triggering condition is a specific condition of a scenario that serves as an initiator for a subsequent system reaction contributing to either a hazardous behavior or an inability to prevent, detect, and mitigate a reasonably foreseeable indirect misuse. [50] Additionally, a reasonably foreseeable direct misuse (such as intentional usage contrary to the manufacturer's or ADS service provider's intentions), which could directly trigger hazardous behavior, is considered a potential triggering condition [50]. Under this definition, the scope of triggering conditions in ISO 21448 encompasses not only environmental conditions but also the ADS itself, by considering possible system usage by the passengers.

Triggering conditions are system-dependent and are decided by system-specific function designs and their insufficiencies. Therefore, a validated triggering condition for one system might have no effect on another system. For example, sun glare may dazzle the camera-based perception of ADS, leading to a system disengagement, while it might not affect ADS that relies on LiDAR for perception. The ISO 21448 standard recommends using scenarios with triggering conditions to conduct vehicle tests and software/hardware in-the-loop tests to evaluate the acceptability of the system's response. Some studies [128, 47] have also highlighted the importance of triggering conditions in generating potentially critical scenarios for scenario-based testing. Thus, triggering conditions are considered as a component of potential critical scenarios to identify whether specific functional insufficiencies exist. Figure 3.1 illustrates the causal model of triggering conditions and their role in testing.

3.5. Hazardous Behavior, Hazard, and Hazardous Event

Hazardous behavior is mentioned in ISO 21448 for interpreting triggering conditions and other terms. However, the term itself lacks a formal definition. Thus, we refer to the relevant notes and illustrations in ISO 21448 and summarize hazardous behavior as *an external, observable, and unintended vehicle behavior*. The presence of hazardous behavior is a criterion to differ triggering conditions from other nominal conditions in a scenario according to the definition in Section 3.4. Absence or presence of hazardous behaviors can be considered as pass/fail criteria in the tests of a black-box system due to their external observability.

Two relevant terms to hazardous behavior are discussed here for clarity, namely hazard and hazardous event. Hazard is defined in ISO 26262 as a potential source of harm caused by the malfunctioning behavior of the item. In the context of ISO 21448, hazards are caused by the hazardous behavior due to triggering conditions and functional insufficiencies.



Example for Hazard: *Rear Collision*
Ego vehicle drives towards **a bridge-like gantry**(1). Due to **a flawed classification algorithm**(2) the gantry is mistakenly recognized as a huge wall. Therefore, the ego vehicle starts **unintended braking**(3), while **another vehicle is following ego vehicle closely**(4).(5) The following vehicle fails to brake in time to control the situation and a rear collision happens. As result, both **vehicles are damaged with driver injured**(6).

Figure 3.1.: Illustration of triggering condition, functional insufficiency, hazardous behavior/event, and harm (©2022 IEEE)

A hazardous event is originally defined in ISO 26262 as *the combination of a hazard and a specific operational situation* [51]. While ISO 21448 does not provide an explicit definition of hazardous event, it is nonetheless beneficial to clarify the relationship among hazard, hazardous event, and hazardous behavior by adapting the definition from ISO 26262 to suit the context of ISO 21448. To this end, we refine the definition of hazardous event as *an incident where a hazardous behavior by the ego vehicle brings a hazard into reality in conjunction with other scenario conditions.*

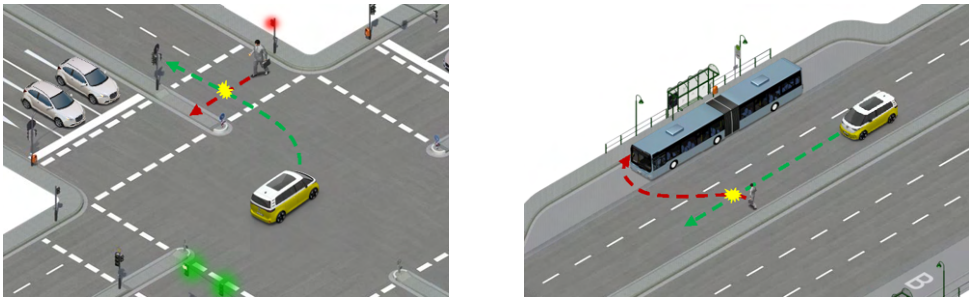
3.6. Formal Definition of Difficult Environmental Condition

The attribute ”difficult” is relative to the capabilities of a system. In the scope of this thesis, the system refers to ADS. In this regard, the concept of *triggering condition* from ISO 21448 has the same consideration as difficult environmental conditions, as explained in Section 3.4. To focus on the external environment, we adopt the concept of triggering conditions while excluding the consideration of misuse of the ADS. Therefore, a difficult environmental condition is defined as *a **condition** in the operational environment of ADS that can trigger one or multiple functional insufficiencies and further result in hazardous behavior of the ADS.*

Based on this definition, three important notes follow: First, a difficult environmental condition is not a scenario but rather a component of a scenario. More specifically, difficult environmental conditions are crucial elements of critical scenarios. Second, difficult environmental conditions do not necessarily correspond to extreme values of parameters within a scenario. The key factors that distinguish difficult environmental conditions from other scenario conditions are their ability to challenge the system's capacity by activating functional insufficiencies and causing subsequent hazardous behavior of the ADS. Lastly, difficult environmental conditions can be downgraded to nominal (i.e., not difficult) environmental conditions through the development and improvement of the ADS.

4. Formalization of Difficult Environmental Conditions[1]

To properly manage large amounts of identification results from different stakeholders is non-trivial. As Cruz et al. [20] point out, vagueness in requirements can cause long-term issues in maintenance and collaboration if stakeholders do not share a common understanding. Additionally, difficult environmental conditions should be integrated into test cases later, while they are often not limited to one scenario context. For example, a difficult environmental condition like a jaywalker can apply to scenarios where the ego vehicle crosses a junction or travels along a straight lane (cf. Figure 4.1). For efficient management and utilization, the difficult environ-



(a) Scenario a: Pedestrian jaywalks when ego performs a protected left turn

(b) Scenario b: Pedestrian jaywalks when the ego follows a straight lane

Figure 4.1.: Different application of the difficult environmental condition *jaywalker* (©2024 Springer)

mental conditions should be documented in a way that avoids vagueness and minimizes the effort needed to integrate them into test cases. In this context, it is crucial to structure difficult environmental conditions with clear formulations and category tags, enabling automation mechanisms for sorting, searching, and mapping them into appropriate scenarios. This chapter introduces standardized descriptions of difficult environmental conditions, which are machine-readable and allow for quick look-up based on specific criteria, e.g., conditions only relevant to vulnerable road users (VRUs).

4.1. Case Study Setup

To develop a general formulation, we start with a simple case study: Firstly, a list of sample difficult environmental conditions was collected from expert interviews and relevant litera-

[1]This chapter is based on Paper I [132] and Paper III [133] and therefore contains verbatim content previously published (©2022 IEEE, ©2024 Springer Nature).

ture [69, 107, 126, 19, 12, 29, 78, 105, 84, 66, 79, 85]. As shown in Table 4.1, these difficult environmental conditions all involve at least one environmental factor (e.g., *tree branch*). Some of these conditions underline a specific attribute (e.g., *with right of way*) or the position (e.g., *from blind area*) of the environmental factor, while the other describe specific events or actions carried out by the environmental factors (e.g., *block the road*). Based on these observations, we conclude three ways how environmental factors can constitute a difficult environmental condition, yielding three types of difficult environmental conditions:

Type I. The presence or absence of a specific environmental factor

Type II. A behavior of an environmental factor

Type III. The interaction of multiple environmental factors

Table 4.1.: Analyzed examples of difficult environmental condition (DEC) (ROW = right of way)

DEC_1	VRU groups at the junctions	DEC_{11}	Wi-Fi signal interferes V2X signal
DEC_2	School bus blocks the road	DEC_{12}	Vehicles with the ROW comes from blind area
DEC_3	Kid jumps out behind the car	DEC_{13}	Concrete barrier in the road color
DEC_4	A tractor in front loses hay	DEC_{14}	Election poster on a crosswalk
DEC_5	Beacon lies on the lane	DEC_{15}	Construction blocks the road
DEC_6	Faded lane marks	DEC_{16}	Debris on the road
DEC_7	Tree branch hanging over the lane	DEC_{17}	An ambulance requires the right of way
DEC_8	Stickers cover the traffic sign	DEC_{18}	Pedestrian provokes the ego vehicle
DEC_9	Bird group on the road	DEC_{19}	Vehicle covered by foam in the parking lot
DEC_{10}	Snow covers the road		

Designing one formalization for all difficult environmental conditions is challenging. Thus, the basic idea is to decompose the problem by proposing formulations for each aforementioned type. To achieve this goal, domain experts are invited to first categorize each exemplary difficult environmental condition from Table 4.1 into one of the concluded types (Type I, II, or III). Afterwards, experts propose formulations for each type by analyzing requirements from management and linguistic aspects. Finally, every difficult environmental condition is rephrased according to the type-based formulations to finalize the standardization. In the following section, the analysis for proposing type-based formulations is described.

4.2. Description Principles and Formulations

In the process of defining type-based formulations, three aspects are considered: firstly, there should be management-level requirements to ensure easy maintenance and readability in difficult environmental conditions. Secondly, basic linguistic concepts should be referred to, so that completeness is checked and the structure is defined for the formulations. Lastly, vocabulary should be restricted, aiming at minimizing unnecessary variances to avoid redundant expressions for one difficult environmental condition. These three aspects are elaborated in the next sections.

4.2.1. Management Requirements

Difficult environmental conditions can be identified and used by different stakeholders. It is necessary to define high-level requirements (so-called management requirements) to ensure correct understanding and easy usage of these conditions. Here, we consider two principles:

(a) Conciseness and Essentials. A difficult environmental condition is identified from scenarios. To simplify the integration of an identified difficult environmental condition into different test scenarios, only essential information is to be documented. For the three types introduced in Section 4.1, the mandatory description units for the documentation are correspondingly (I) the environmental factor, (II) the environmental factor and its behavior, and (III) the environmental factors and their interaction.

- Environmental Factor (EF) can be a tangible entity (e.g., road users), an intangible condition (e.g., illumination), or a signal (e.g., Wi-Fi)
- Behavior (BHV) is an action (e.g., braking) or a transition of states (e.g., traffic light turns red) of an EF.
- Interaction (INTR) is either an action conducted by an EF to the other EF (e.g., a parking vehicle occludes a pedestrian) or mutual actions

(b) Intuitiveness. The formulation should reveal the effect of difficult environmental conditions. Thus, we include a link to ADS functions as another mandatory description unit.

- Link to Function (L2F) represents expert opinions on a potentially impacted function by a given difficult environmental condition. L2F serves as a direct index to sort difficult environmental conditions for functional-oriented tests. It also provides a direction of investigation when the system response is deemed unacceptable in a scenario.

For the sake of simplicity, we propose three super-types for L2F, namely sense, plan, and act, which can be refined into corresponding sub-functions (e.g, localization) or components (e.g., camera). Although the effect of difficult environmental conditions can be propagated through the sense-plan-act chain and thus expose multiple functional insufficiencies, we suggest documenting only the first encountered insufficiency in the chain. Considering *a plastic bag flies above the path* causing an emergency braking of the ego vehicle, firstly, the ego vehicle fails to classify the plastic bag, then it decides to brake in front of the unknown object. In this case, although the planner could make better decisions (e.g., performing a lane change), the perception is to blame first. If the plastic bag would be recognized correctly, the probability of a hazardous planning decision is rather low. Therefore, the recommended L2F is *[sense]*.

Sometimes, merely using mandatory description units is not enough to fully reveal the effect. For instance, the ADS drive at night and the camera fails to detect a pedestrian, because the pedestrian wears a dark jacket. While the presence of the pedestrian is the primary trigger for the misdetection, describing the difficult environmental condition as just *a pedestrian [sense]* can barely reveal the issue. A more complete and self-explanatory description would be *a pedestrian in dark jacket at night [sense]* for documenting the problem. Hence, we introduce two additional optional description units to include the necessary attributes and temporospatial context of the relevant environmental factor.

- Attribute (ATTR) is a status, a feature, or the right of way (if the related EF is a road user) of an EF, which remains unchanged or valid in the scenario context.

- Temporospatial Context (TSC) is the scenario context describing the timing, duration, and location of the existence. It can also describe the behavior or interaction of the relevant EFs. TSC can also be adverbs of degree (like fully, partially, and drastically) denoting the extent of a behavior or an interaction.

4.2.2. Linguistics Consideration

After determining the description units, they should be assembled for establishing the type-based formulations. The formulations are comparable to condensed and structured sentences or parts of sentences in natural language. Some basic concepts from linguistics are therefore helpful to verify the completeness of the description units and determine their connections. For example, we consider the five essential elements of a sentence, namely subject, predicate, object, modifier, and complement[2]. The description units of difficult environmental conditions can be mapped onto these five components as follows:

Environmental factor is equal to the subject or object in a sentence, thus can combine with a predicate, be modified by an attributive modifier, or be linked to a complement.

Behavior is comparable to an intransitive predicate. It describes the action of the subject, which does not require an object to complete the meaning (e.g., *pedestrian jaywalks*).

Interaction is comparable to a transitive predicate. It requires both the subject and the object to complete the meaning of the action (e.g., *a school bus blocks the road*).

Temporospatial context is comparable to an adverbial modifier. It can modify the subject, the object, or the predicate.

Attribute is comparable to an attributive modifier (e.g., *black* vehicle) or a complement (e.g., *right of way* sign), depending on the concrete content. It can modify or be assigned to the subject or the object.

Meanwhile, diverse tenses in linguistics should generally be avoided when describing difficult environmental conditions. This is because such conditions represent a status or event, which can be sufficiently described and understood with the present tense. Similarly, only one voice should be used to prevent redundant descriptions of the same difficult environmental condition. To ensure clarity, the active voice is recommended. For instance, when describing a behavior like 'cover,' 'A covers B' is valid, while 'B is covered by A' is not. Based on these guidelines, we determine the structure of the formulations for each type in Figure 4.2.

[2]These elements can be named differently in literature, e.g., in [95] they are called *subject*, *predicate*, *object*, *description*, and *complement*

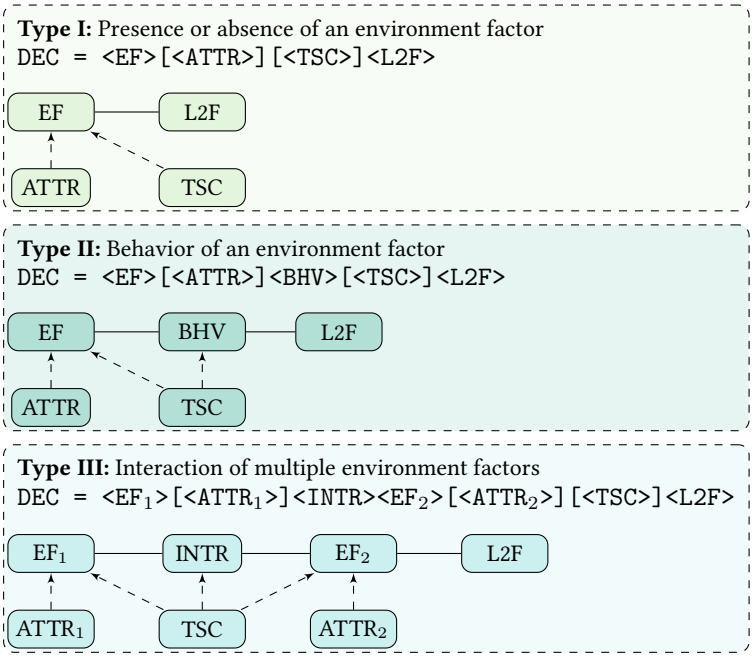


Figure 4.2.: Three types of difficult environmental conditions and their formulation: EF: Environmental Factor, ATTR: Attribute, L2F: Link to Function, TSC: Temporospatial Context, BHV: Behavior, INTR: Interaction. Solid lines connect mandatory elements. Dashed arrows denote the connections between optional and mandatory elements. (©2022 IEEE)

4.2.3. Vocabulary

To further restrict the diversity in the description of difficult environmental conditions, vocabulary can be specified for each description unit. Essentially, environmental factors in difficult environmental conditions are part of the ODD of ADS. Meanwhile, describing an ODD with a taxonomy should be part of the ADS specification process. There, ODD elements are determined, such as *Vehicle* and *Pedestrian*. Therefore, the vocabulary for describing difficult environmental conditions can be directly adopted from the predefined ODD taxonomy. This approach has two main benefits: Firstly, the ODD taxonomy and difficult environmental conditions are both important working artifacts throughout the ADS development and verification cycle. Aligning their taxonomies and vocabulary enhances consistency and traceability in design and testing requirements, simplifying communication across departments and stakeholders, contributing to work efficiency. Secondly, formalized difficult environmental conditions can be mapped to the ODD, indicating which ODD elements are covered by the identified difficult environmental condition. The mapping result can be used for deriving coverage metrics to evaluate the completeness of the identification. At the same time, conditions that can not be mapped to the ODD can reveal potential insufficiencies in the ODD specification, therefore contributing to improving the ADS specification.

Figure 4.3 illustrates a vocabulary tree for describing difficult environmental conditions, derived from a reference ODD taxonomy. The ODD elements are organized according to a five-

layer scenario model [36]. Each environmental factor can be characterized by Behavior or Interaction (BHV or INTR), Attribute (ATTR), and Temporospatial Context (TSC), forming Type I/II/III difficult environmental conditions. Accordingly, there are simplified, exemplar lists of vocabulary for describing BHV/INTR, ATTR, and TSC. For instance, to describe briefly and clearly that a vehicle in front of the ego vehicle starts to reverse, the most relevant vocabulary from the tree are *Vehicle*, *Leading*, and *Reverse*.

4.3. Further Systematization Possibilities

Additional category labels can facilitate searching for suitable difficult environmental conditions for scenario-based tests. We consider the categorization based on diverse criteria.

(a) Environmental factor origin. The origin of involved environmental factors helps to clarify if and where a difficult environmental condition can be injected into a given scenario. On the macroscopic level, possible environmental factors vary among different application domains like urban, rural areas, and highways. E.g., wild animals are unlikely to appear in urban areas, while cyclists are not included in highway scenarios. On the microscopic level, an environmental factor belongs to a specific scenario layer (cf. scenario layer model [36, 97]).

(b) Linked functional insufficiency. Since the test aims to investigate the vulnerability of an automated vehicle, it is practical to index difficult environmental conditions by their links to functions. For instance, in order to verify the perception performance, all difficult environmental conditions with links to the perception component are relevant. Besides denoting the Link to Function (L2F) in the formulation, a more detailed functional decomposition model (cf. [5]) for ADS can be utilized as a categorization criterion.

(c) Occurrence frequency. When exhaustive testing becomes unrealistic due to the vast number of difficult environmental conditions, arguments for prioritizing the candidates are vital. One common and system-agnostic prioritization criterion is the occurrence frequency. It is natural to consider those often occurring difficult environmental conditions as more interesting test candidates than those with extremely low likelihood to be encountered during the operation. Based on the frequency, difficult environmental conditions can be divided into three groups as *Common Case*, *Reported Case*, and *Hypothesized Case*. Common cases denote the occurrence on a daily basis (e.g., a crowd of pedestrians). Reported cases are validated by media (e.g., news) or mentioned in safety reports by automotive manufacturers, which exhibit lower exposure within normal drives. Hypothesized cases are theoretically possible, but are never encountered in reality.

4.4. Formalization Results

Following the aforementioned requirements and concepts, all examples in Table 4.1 are assigned to a type and are formalized with defined description units, rules, and vocabulary. Table 4.2 illustrates the formalization results: The first two columns denote four categories suggested in Section 4.3. The third column shows the type (I/II/III) for each sample. The following seven columns reserve cells highlighted in different green colors for mandatory and optional elements according to each type. Concrete values are extracted from the original expression

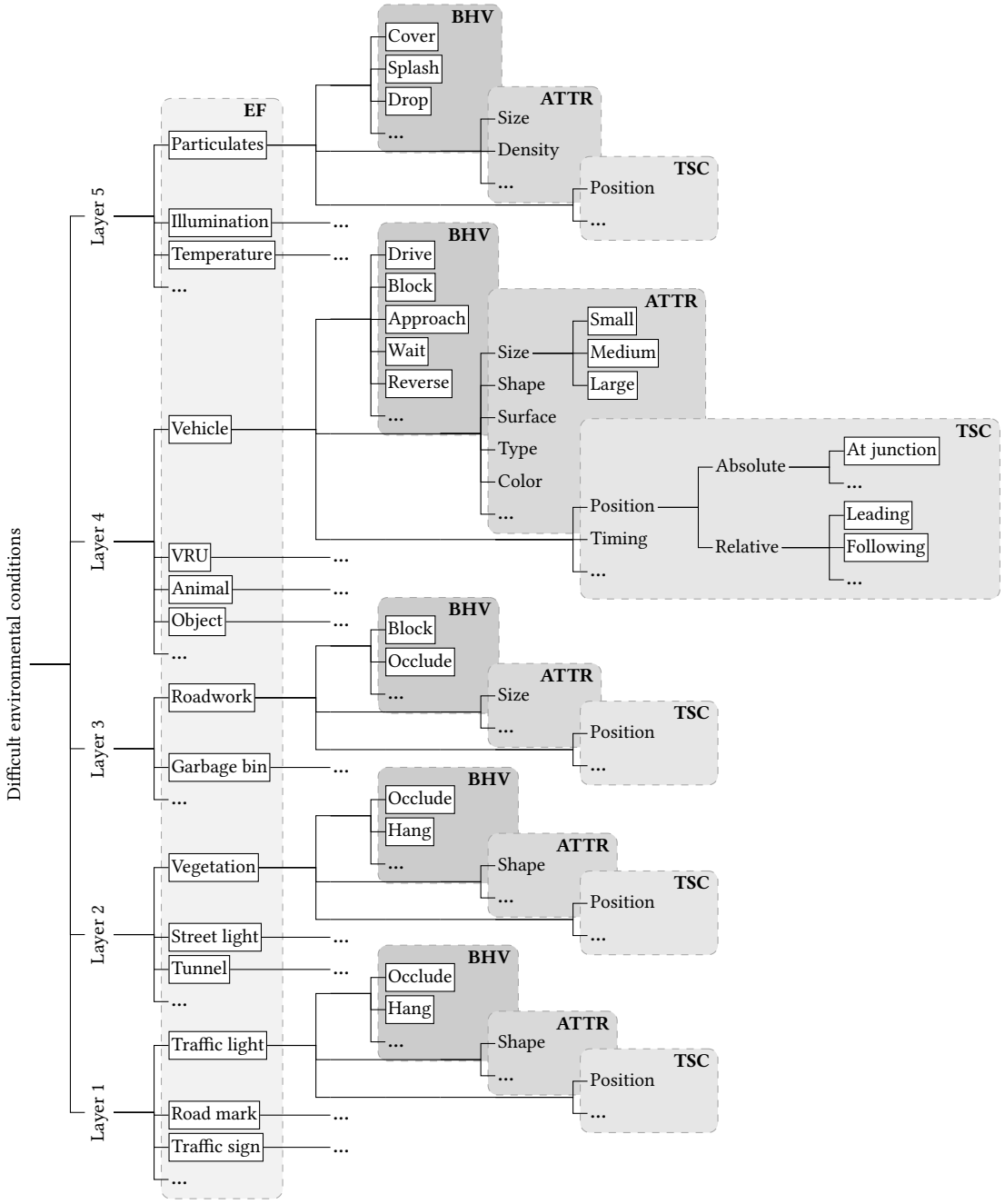


Figure 4.3.: A vocabulary tree for describing difficult environmental conditions

to fit in individual table cells. The last column denotes literature origins for the examples. In the upcoming Chapter 5 and Chapter 6, an experiment is conducted to identify larger amounts of difficult environmental conditions. The identifications are all successfully standardized according to the proposed formalization and are documented in Appendix A, indicating the effectiveness of the formalization method in this chapter.

Table 4.2.: Formalization results (ROW = right of way, VRU = vulnerable road user) (©2022 IEEE)

Cat.	Description			Example							Ref.
				EF	ATTR	BHV	INTR	EF	TSC	L2F	
EF Origin (Domain)	Urban	Downtown	I	VRU group					at the junction	sense	
		School area	III	School bus			blocks	road		plan	[107]
		Residential	II	Kid		jumps out			behind car	plan	[126]
	Rural		II	Tractor		loses hay			in front	sense	
	Highway		II	Beacon		lies			on the lane	sense	[19]
EF Origin (Scenario)	**L1 Road network**		I	Lane marks	faded					sense	[12]
	L2 Roadside		I	Tree branch					over lane	plan	
	L3 Temp. Mod.		III	Sticker			covers	a traffic sign	partially	sense	[69]
	L4 Dyn. objects		I	Birds group					on road	plan	
	L5 Env. condition		III	Snow			covers	road		act	[29]
	L6 Digital inf.		III	Wi-Fi signal			interferes	V2X signal		sense	[78]
Linked FI	**Inf. Access**		II	Vehicle	with ROW	comes			from blind area	plan	[105]
	Inf. Reception		I	Concrete barrier	in road color					sense	
	Inf. Processing		I	Election poster					on crosswalk	sense	
	Understanding		III	Construction			blocks	path		plan	[84]
	Action		I	Debris					on road	act	[66]
Fre-quency	Common Case		II	Ambulance		requires ROW				plan	[79]
	Reported Case		III	Pedestrian			provokes	ego		plan	
	Hypothesized		I	Vehicle	in foam				in parking lot	sense	[85]

Part III.

Elicitation of Difficult Environmental Conditions and Related Scenarios

5. Knowledge-driven Approach to Identify Difficult Environmental Conditions[1]

Expert knowledge is a crucial and indispensable input for comprehensively identifying difficult environmental conditions, particularly those that may impact a wide range of ADS implementations. While many existing studies address this topic, they often focus narrowly on specific components or functions of ADS. A dedicated method for systematically integrating expert knowledge at the system level, however, remains lacking. In this chapter, we introduce a designated, knowledge-driven method for eliciting difficult environmental conditions. A case study based on an urban scenario is presented to demonstrate the application of the method. Finally, we compare our method with other knowledge-driven approaches that could potentially be used for the same research purpose and discuss the capabilities and limitations of our method.

5.1. Scenario-based Hazard and Fault Analysis

With an understanding of how difficult environmental conditions contribute to hazardous behaviors, a customized knowledge-driven method called Scenario-based Hazard and Fault Analysis (SHFA) is designed to identify potential difficult environmental conditions from scenarios or driving data. The SHFA method analyzes scenarios in three steps (cf. Fig. 5.1): (1) Scenarios are modelled as *intended maneuvers* of ADS and *relevant scenario elements*. (2) Hazardous maneuvers are derived from intended maneuvers with a look-up table. (3) Difficult environmental conditions for each hazardous maneuver are iteratively identified by analyzing relevant scenario elements and the ADS components in a concrete situation. The following sections introduce SHFA in detail.

5.1.1. Step One: Modelling Scenario

A scenario is a temporal sequence of actions/events and scenes [114]. Actions and events can be elicited based on the ego vehicle's intention. Scenes are snapshots of the environment and can be decomposed to ODD elements. Thus, scenarios modelling is divided into eliciting ego vehicle intention and scenario elements.

Intended ego vehicle maneuver. To describe ego vehicle's intentions, we selectively adopt a vehicular maneuver taxonomy [44] as atomic units and divide them into three groups: vehicle state maneuvers (velocity changes or preservations), lane-related maneuvers (relative position

[1]This chapter is based on Paper I [132] and Paper III [133], and therefore contains verbatim content previously published (©2024 Springer Nature).

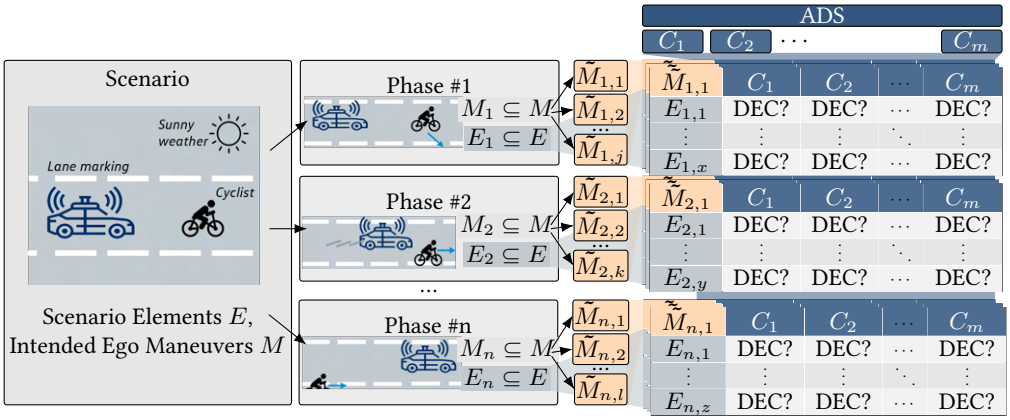


Figure 5.1.: SHFA Method Overview: ADS components (C), hazardous maneuver ($\tilde{M}$), difficult environmental condition (DEC) (©2024 Springer Nature)

to lanes), and junction-related maneuvers (activities on junctions). Two implicit maneuvers *Enter junction* and *Exit junction* are proposed additionally as context for junction-related maneuvers. Maneuvers from the same group are mutually exclusive, but can transition from one to another (cf. Fig. 5.2). Maneuvers from different groups can be performed concurrently and result in a Parallel Maneuver Combination (PMC) [44]. Based on the PMCs, a scenario is segmented into chronological phases.

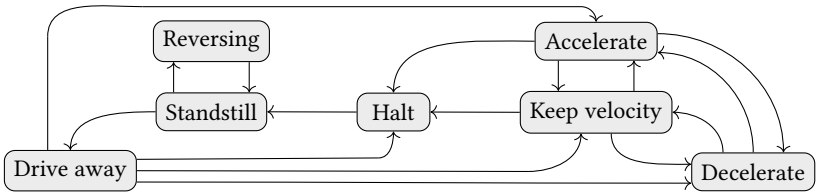
Scenario elements. Difficult environmental conditions consist of scenario elements. To thoroughly explore a scenario, it is vital to enumerate its scenario elements exhaustively. First, all scenario elements are grouped based on a six-layer scenario model [97] into road network and traffic guidance objects, roadside structures, temporary modifications, dynamic objects, environment conditions, and digital information. Then, the relevancy of scenario elements is evaluated considering specific criteria, e.g., based on interactions with the ego vehicle [82].

5.1.2. Step Two: Deriving Hazardous Maneuvers

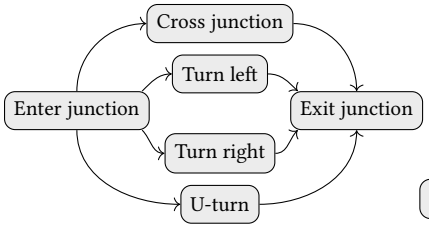
With identified intended maneuvers, SHFA deduces all possibilities of how the ego vehicle can perform unintended and hazardous maneuvers instead.

Unintended maneuvers. Unintended maneuvers occur if intended maneuvers are not executed, improperly executed regarding timing, extent, and duration, or completely falsified by another maneuver. We refer to a set of guide words from the HAZOP method [18] (including *not provided*, *too late*, *too early*, *too much*, *too little*, *too long*, *too short*, and *falsified*) and combine them with intended maneuvers for deriving unintended maneuvers. To avoid ambiguity or redundancy during the combination, SHFA provides a look-up table and additional rules. Thus, given an intended maneuver, a minimal set of explicit and meaningful unintended maneuvers can be looked up from Table 5.1, considering:

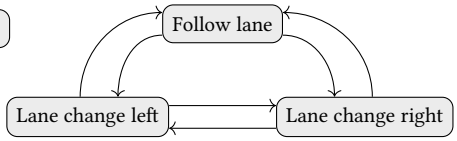
Rule 1. By analyzing a maneuver in a certain phase of the scenario, we assume that the intended maneuver from the previous phase is correctly performed.



(a) Vehicle state maneuvers: Standstill, Halt & Drive away are special cases of Keep velocity, Decelerate & Accelerate respectively. They are defined separately.



(b) Junction-related maneuvers



(c) Lane-related maneuvers

Figure 5.2.: Maneuver transition charts: Detailed transitions refer to Hartjen [44]. (©2024 Springer Nature)

Rule 2. The guide word *falsified* signifies a maneuver different from the current and the last intended maneuver. The falsified maneuver should be able to transition from the intended maneuver of the previous phase (cf. Fig. 5.2).

While *Rule 1* keeps the analysis simple, *Rule 2* guarantees that the *falsified* maneuver is reasonable in the scenario context and different from *not provided*.

Hazardous maneuvers. To identify hazardous maneuvers, the compatibility and consequence of derived unintended maneuvers are evaluated in concrete situations. For example, an intended *Follow lane* maneuver can be *falsified* to *Lane change right* and *Lane change left*. If the ego vehicle is already in the rightmost lane, *Lane change right* is irrelevant. Besides, an unintended maneuver is deemed hazardous if it violates traffic rules (e.g., crossing a stop line).

5.1.3. Step Three: Eliciting Difficult Environmental Conditions

With the modeled scenario and identified hazardous maneuvers, experts iteratively evaluate their cause-and-effect relations to identify difficult environmental conditions.

Heuristic matrix. A heuristic matrix organizes an exhaustive identification process for each hazardous maneuver. One dimension lists all scenario elements relevant to a scenario phase. The other dimension lists the ADS components along the sense-plan-act chain. With a white-box system whose detailed implementation is available, concrete algorithms and functions should be considered for finding more concrete difficult environmental conditions. Each entry in the matrix represents a *unit analysis* based on a scenario element and a specific function.

Unit analysis. For each unit analysis, experts evaluate whether the presence, absence or characteristics of the scenario element could lead to insufficiency of the component and result in the current hazardous maneuver. Once a causal-effect relation is established, a difficult environmental condition is registered in the corresponding cell of the heuristic matrix. Exemplarily, faded lane marks could lead to inaccurate lane mark detection and trigger the ADS to deviate

Table 5.1.: Look-up table (©2024 Springer Nature)

Keyword	**Intended Maneuver**											
	Stand-still	Reversing	Drive-away	Accelerate	Decelerate	Halt	Keep vel.	Follow lane	Change {l./r.}	Turn {l./r.}	U-turn	Cross jct.
¬ provided		Standstill		Keep preceding maneuver[a]								
Too late	Start maneuver too late[a]									Start maneuver after it was required		
Too early	Start maneuver too early[a]									Start maneuver despite waiting required		
Too much		Reverse too far	Abrupt start	Acceleration or duration exceeded		Drastic stop			Exceed target lane			
Too little		Reverse too few	Start not in time	Acceleration or duration not reached		Halt not in time		Deviate from lane	Lane not reached			
Too long							Subsequent maneuver not in time[b]		Maneuver unnecessarily slow			
Too short							Subsequent maneuver too soon[b]		Drastic steering {l./r.}			
Falsified	Drive-away, Reversing	Drive-away	Reversing	Keep vel., Dec., Halt	Keep vel., Acc., Halt	Keep vel., Acc., Dec.	Acc., Halt, Dec.	Lane change {l./r.}	Follow lane, Change {l./r.}	Any other unintended junction maneuver is performed		

a: Applicable if the maneuver is not the first maneuver of the scenario; b: Applicable if the corresponding subsequent maneuver is available

from lane (cf. Table 5.1: *Follow Lane* × *Too little*). Features of scenario elements can be adapted during the evaluation. However, adaptations shall not change the substantial dynamics of the original scenario (e.g., by assuming a completely different behavior of the scenario element), so that the intended ego maneuvers would be different.

5.2. Illustrative Example

To illustrate the method, we analyze one scenario (cf. Fig. 5.3). The scenario is set on a road with sidewalks and two lanes for two directions, divided by a solid lane marking. The road is partially elevated by a bridge. The ego vehicle follows the lane on the right and passes two cyclists sequentially, adapting its velocity. Several oncoming vehicles pass the ego vehicle on the left. A vehicle drives behind the ego vehicle.

5.2.1. Modelling Scenario

According to the ego vehicle's interactions with the two cyclists over time, the scenario can be divided into three phases with intended PMCs:

Figure 5.3.: Video images of the scenario (©2024 Springer Nature)

- Phase #1 [*Follow lane*, *Keep velocity*]: the ego vehicle drives past *Cyclist #1* with constant velocity, as the cyclist pauses on the lane edge.
- Phase #2 [*Follow lane*, *Decelerate*]: the ego vehicle approaches *Cyclist #2* with decreased velocity to maintain the safe distance to the cyclist.
- Phase #3 [*Follow lane*, *Accelerate*]: the ego vehicle drives past *Cyclist #2* with increased velocity, as the road becomes broader.

We only illustrate Phase #1 for the following steps. Relevant scenario elements are observed and categorized according to six scenario layers (cf. Table 5.2a).

5.2.2. Deriving Hazardous Maneuver

Intended maneuvers in Phase #1 are *Follow lane* and *Keep velocity*. As they are the first maneuvers of the scenario, guide words like *not provided*, *too early*, and *too late* are irrelevant. With the remaining guide words, unintended maneuvers are looked up from Table 5.1 and registered in Table 5.2b. Among them, *Lane change right* is incompatible to the given road network. *Accelerate* and *Decelerate* are not hazardous due to sufficient lateral distance to Cyclist #1. *Deviate from lane*, *Lane change left* and *Halt* are hazardous for violating safe distances to the oncoming and following traffic.

5.2.3. Eliciting Difficult Environmental Conditions

The heuristic matrix is arranged by relevant scenario elements of Phase #1 and the decomposed ADS. Without a reference system, we refer to a function decomposition model [5] and generally decompose ADS into five stages, namely information access, information reception, information processing, plan and action, while the first three stages belong to sense components (cf. Table 5.2c).

We illustrate the identification of the hazardous maneuver *Halt*. The unit analyses are conducted within the matrix from left to right and from top to bottom. During the unit analyses of each scenario element, experts can adapt certain static features (e.g., appearance) or local behavior (e.g., car activating turn indicator) of the scenario element. In this way, the scenario keeps its essence, while the counter-factual adaptations enable experts to identify more difficult environmental conditions. We differentiate whether a scenario element is adapted (or not) during the analysis by the wording *assume* and *confirm* respectively:

Table 5.2.: Analysis of Scenario Phase #1 (©2024 Springer Nature)

(a) Relevant elements

Scenario layer	Relevant scene elements
L1. Road Net.	Lane, solid lane marking, curb
L2. Roadside	Trees, barriers, grass
L3. Temp. Mod.	n/a
L4. Dynamic objects	Cyclist #1, oncoming vehicle, pedestrians, parking bikes
L5. Env. Cond.	Cloudy weather
L6. Digital Inf.	n/a

(b) Unintended/**Hazardous** maneuvers

Guide word	Intended maneuver	
	Keep velocity	Follow lane
Too little		**Deviate from lane**[1]
Too long	[Phase #2] Decelerate too late	
Too short	[Phase #2] Decelerate too soon	
Falsified	Accelerate / Decelerate / **Halt**[2]	**Lane change left**[3] /Lane change right

(c) Identified difficult environmental conditions (Superscripts refer to corresponding maneuvers in Table 5.2b)

Scenario element	Functional layers				
	Information access	Information reception	Information processing	Plan	Action
Lane		[2]steep slope	[2]unregistered slope		
Solid lane marking	[1]faded				
Curb	[1]grass covers curbs				
Grass			[1]tall grass around roadside barrier		
Barriers					
Cyclist #1		[2]clothing with low contrast to ground	[2]close proximity to roadside barrier, [2,3]unusual size/shape	[2]halts on the road edge with hand gestures	
Other elements					

- *Lane*×*information reception*: experts confirm that an elevated lane/a slope can lead to multi-path effects of Radar sensors and thus cause the detection of phantom objects and finally lead to an unintended *Halt* maneuver.
- *Lane*×*information processing*: experts confirm that an unregistered slope geometry can lead to the detection of floating objects with the wrong position.
- *Cyclist #1*×*information reception*: experts assume that if the cyclist's clothing has low contrast to the background, it could be detected too late by cameras.
- *Cyclist #1*×*information processing*: experts confirm that the cyclist touching a bike rack on the roadside could impede object segmentation and even be detected as a cyclist or a vehicle in a traversed direction on the lane, which can confuse the planning component to require an emergent braking.
- *Cyclist #1*×*information processing*: experts assume that, in general, if a cyclist has an unusual shape or size, object detection could fail or be delayed.
- *Cyclist #1*×*plan*: experts assume that possible hand gestures of the cyclist could be misinterpreted as an intention to return to the lane center.

Similarly, *Deviate from lane* and *Lane change left* are analyzed in their own matrix. We fuse all identified difficult environmental conditions from Phase #1 in Table 5.2c.

5.3. Capabilities and Limitations

In this section, we compare the SHFA method with other established or state-of-the-art knowledge-driven methods with similar applications, providing a theoretical argument for the capabilities of SHFA. Following this, we discuss the limitations of SHFA.

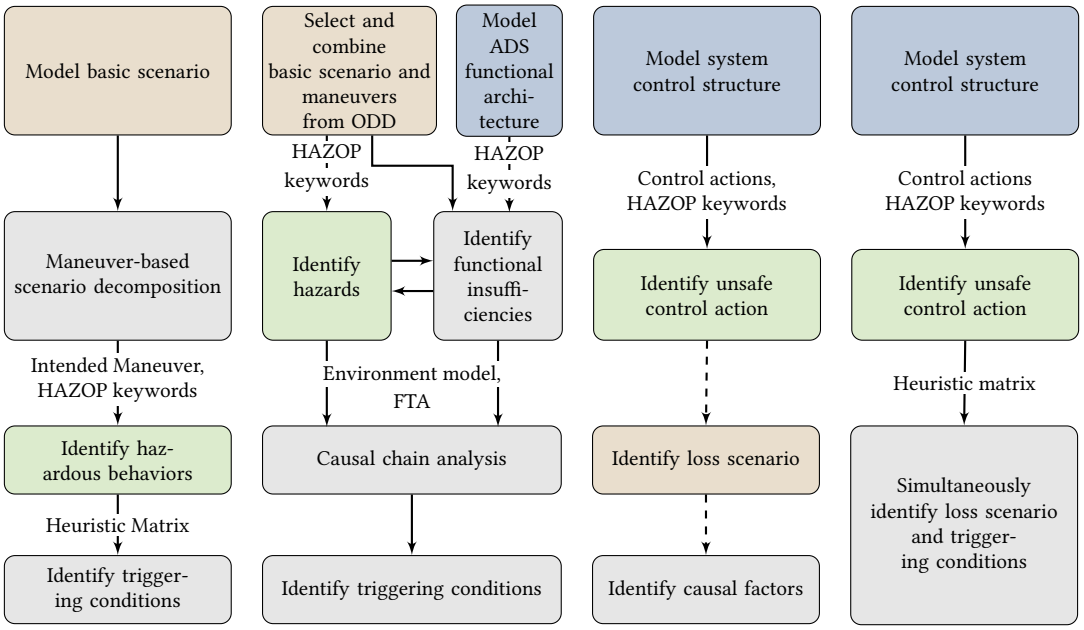
5.3.1. Comparison to Other Knowledge-driven Methods

To compare the SHFA method with other existing knowledge-driven methods, we first investigate established analysis techniques in the automotive safety domain. These include the deductive, top-down Fault-Tree-Analysis (FTA) [49], the inductive, bottom-up Failure Mode and Effects Analysis (FMEA) [48], and the explorative, top-down System Theoretic Process Analysis (STPA) [68].

For example, ISO 21448 [50, B.3.2] demonstrates the use of Cause-Tree-Analysis (CTA), which is similar to Fault-Tree-Analysis (FTA), to model the dependencies between hazards (e.g., sudden deceleration), functional insufficiencies (e.g., an object falsely classified as an obstacle), and triggering conditions (e.g., a large, low-hanging tree branch overhanging the roadway). While this enables the calculation of minimal cut sets that help assess risk mitigation measures, the method does not contribute to identifying triggering conditions (or difficult environmental conditions in this thesis) but builds on already identified triggering conditions.

Additionally, the ISO 21448 [50, B.3.3] presents an inductive method analogous to FMEA, where a worksheet is continuously filled out to capture failure modes and system-level effects. In this case, the inductive method breaks down system elements and their functional insufficiencies, describing their relations to potential triggering conditions and resulting hazardous behaviors. It also outlines measures to address the insufficiency and provides a rationale for acceptance. Like the deductive method, the inductive method facilitates traceability between functional insufficiencies, triggering conditions, and hazardous behaviors, but it lacks guidance on how to identify triggering conditions initially. Therefore, we exclude FMEA, FTA, CTA, and other variants of these methods for identification purposes.

STPA is also mentioned in ISO 21448. Unlike the aforementioned methods, STPA is explicitly recommended for identifying triggering conditions. STPA begins with an analysis of the system control structure and ends with identifying loss scenarios, where triggering conditions can be further derived. Therefore, the STPA method holds potential for identifying difficult environmental conditions in this thesis. Similarly, a state-of-the-art method by Kramer et al. [67] for identifying hazardous scenarios is potentially helpful in identifying difficult environmental conditions. Consequently, we further compare the SHFA method with the STPA method and the method by Kramer et al. in detail. Figure 5.4 illustrates the process of each method. Similar procedures are presented in the same color. We evaluate the three methods based on three aspects: first, the cost of initializing the process (e.g., required input and knowledge); second, the consistency of the procedures to ensure ease of execution and traceability of intermediate



(a) Our method (b) Method by Kramer et al. [67] (c) STPA [68] (d) STPA with our heuristic matrix

Figure 5.4.: Overview of diverse identification methods: the term *triggering condition* is equal to *difficult environmental condition* in this thesis (©2022 IEEE)

results; and third, the diversity of the identified difficult environmental conditions to ensure comprehensive test coverage.

The method by Kramer et al. initially aims to identify and quantify hazardous scenarios. Environmental conditions that can trigger the hazards are derived in the first several steps. The method starts with modeling the functional architecture of a non-black-box system. Such a system model is typically not available at the early stage of development for multiple reasons. Then, hazards and functional insufficiencies as well as a local causal-effect chain are derived with traceability using established HARA techniques. A further global causal-effect analysis with the assistance of an environment model for expressing difficult environmental conditions can be necessary for a more holistic identification. The adopted techniques (e.g. HAZOP-inspired keywords and Fault Tree Analysis) provide theoretical guidance for continuing with the procedures. However, the work lacks a coherent case study to demonstrate the whole process of the method.

The STPA method is first proposed for identifying safety constraints and requirements for complex systems. Firstly, a control structure is modeled, including the system, the driver, the environment and the interactions among them. The commands among system functions or between the driver and a specific function module are enumerated as control actions. Subsequently, unsafe control actions and hazards are derived with HAZOP-like keywords. The interaction between the environment and the system is described as an influence. However, the analysis of influences is not further presented in the method. Next, loss scenarios need to be hypothesized according to each unsafe control action, so that causal factors, including

functional insufficiencies and difficult environmental conditions, can be identified. Two major drawbacks are evident in the STPA method for identifying difficult environmental conditions. Firstly, it lacks sufficient guidance for building up the system control structure. To what extent the vehicle functions should be modeled and how the environment should be modeled remain unclear. Secondly, a necessary structure is missing to identify loss scenarios from unsafe control actions and causal factors from loss scenarios. For example, there could be countless scenarios where *a desired deceleration command is not provided* (unsafe control action). The experts have to randomly brainstorm a subset of such scenarios without any other supportive information. As a result, the identified loss scenarios and causal factors could be limited in their number and exhibit certain similarity. An additional step of systematically identifying hazardous behaviors can be constructive to fill the gap, as pointed out by Graubohm et al. [40].

The SHFA method is designated for the identification of difficult environmental conditions for ADS. Different from the method by Kramer et al. and the STPA method, the analytical method SHFA is driven by scanning driving scenarios and identifying potential elements of interest. By means of a comprehensive framework, SHFA utilizes models to exhaustively decompose the environment and the ADS, and deduce hazardous maneuvers. These decomposition procedures are free of human bias and show potential for automation, since they follow defined taxonomies (e.g., the description of maneuvers and scenario layers) and finite description units (e.g., the countable amount of general unintended maneuvers). Exemplarily, the concept of relevant areas by Philipp et al. [82] can help to automatically associate scenario elements with ego maneuvers. A non-black-box system is helpful to identify concrete, system-specific difficult environmental conditions. However, it is not a crucial requirement for applying the method. In this regard, SHFA is especially advantageous in efficiently identifying system-agnostic difficult environmental conditions for analysis and testing purposes in early development phases, when there is a lack of field-testing data or when the detailed system knowledge is not accessible.

5.3.2. Limitations of the SHFA Method

The designated SHFA method supports the systematic and efficient identification of difficult environmental conditions and offers advantages compared to other non-designated methods. However, we are aware of two main limitations of the SHFA method:

Completeness and reasonable risk for ODDs. The parameter space of an open world is infinite, and so are the possible environment constellations. Although SHFA is capable of identifying large amounts of potential difficult environmental conditions with relatively small experiment resources, we are aware of the fact that there can exist many more undiscovered factors. At the end, a proof cannot be made that all potential difficult environmental conditions are identified with SHFA from an open world. Therefore, it is essential to define a criterion for sufficiency of the identification, instead of proving completeness. In the context of ISO 21448, the sufficiency can be shown by reasonably low residual risk. In practice, SHFA analyzes scenarios that are associated with ODD elements. Sufficiency of difficult environmental condition identification with SHFA could be estimated and approached by considering coverage of the predefined ODD via the analyzed scenarios. To that end, the scenarios can be defined in accordance with the requirements in the standard UL 4600 [113] and the concept of nominal scenarios in the regulation EU 2022/1426 [27].

Bias via human factors. Expert knowledge plays a dominant role in the final step of SHFA, which is to elicit environmental causes for previously identified hazardous maneuvers. On the one hand, we regard the involvement of expert knowledge as a necessary strategy for systematically, comprehensively, and efficiently analyzing the environmental influences for a complicated ADS. On the other hand, the expert analysis for associating scenario elements with specific system insufficiencies is still presented qualitatively in the current SHFA method. In further development of the SHFA, the expert analysis could be assisted and quantified with statistical methods and correlation models (e.g., the implementation of a Bayesian network by Adee et al. [2]).

6. Practice of SHFA Method and Findings[1]

In order to examine the applicability of the SHFA method introduced in Chapter 5 and to establish the first catalog of difficult environmental conditions, four workshops were conducted with 16 domain experts, including function developers and testing engineers in the field of automated driving. As a result, 122 difficult environmental conditions were elicited and further analyzed. In this chapter, we provide an overview of the workshop organization and implementation. Besides, we present the expert feedback from the workshops along with corresponding reflections as a qualitative evaluation of the SHFA method. Finally, we discuss the findings from the analysis of the elicitation, including the formalization, clustering, and distribution of the identified difficult environmental conditions.

6.1. Workshop Design and Implementation

This section outlines the design and implementation of the expert workshops. The workshop process is described across three key dimensions, namely working material, workshop schedule and participants, and actual method implementation and refinement.

Working Material. As the SHFA method takes driving scenarios as input to start the analysis of potential hazards, the major working material for the workshops is a set of scenarios. Considering that the main purposes of conducting the workshops are to practically evaluate the SHFA method and to accumulate an initial set of potential difficult environmental conditions, we select scenarios with a certain level of complexity (to increase the likelihood of identifying difficult environmental conditions), but do not aim at defining a complete set of scenarios (which would aim for comprehensive identification). Thus, the selected scenarios meet two qualitative criteria for complexity: First, there should be at least one object interacting with the ego vehicle. Second, the ego vehicle should exhibit at least two behaviors over time. Principally, scenarios can be synthesized from scratch to fulfill these two criteria. For efficiency, we selected ten scenarios directly from 192 min of video recordings of a manual drive according to the depicted criteria. The video data was captured from the ego vehicle's perspective in urban traffic in Hamburg, Germany.

Workshop Schedule and Participants. The four workshops were conducted over the course of six weeks, with each workshop planned for 2.5 h. A diverse group of ADS function developers (FD) and system testing experts (TE) was invited as participants. The function expertise of the developers included, but was not limited to, perception and planning algorithms, machine learning, sensor integration, and localization. The testing experts specialized in functional safety, simulation, and scenario-based testing. Of all the experts, only Expert #1,

[1] This chapter is partly based on Paper III [133] and therefore contains verbatim content previously published (©2024 Springer Nature).

#6, and #10 had prior knowledge of the SHFA method before attending the workshops, due to previous discussions. In addition to the expert participants, one moderator was assigned to provide an introduction, offer necessary guidance, and manage time during the workshop. The allocation of experts to workshops, workshop dates, analyzed scenarios, and the number of difficult environmental conditions identified are listed in Table 6.1. A general agenda

Table 6.1.: Workshop Overview

WS-ID	Participants & Background		Date	Scenarios	Ident. amount
1	Expert 1	TE	28.07.22	#1, #2, #17	31
	Expert 2	TE			
	Expert 3	TE			
2	Expert 4	FD	12.08.22	#25, #10, #11	26
	Expert 5	FD			
	Expert 6	FD			
3	Expert 7	TE	25.08.22	#21, #7, #27	42
	Expert 8	FD			
	Expert 9	FD			
	Expert 10	FD			
4	Expert 11	FD	08.09.22	#16 + Scenario-agnostic analysis	20 + 43
	Expert 12	FD			
	Expert 13	FD			
	Expert 14	TE			
	Expert 15	TE			
	Expert 16	TE			

is planned for the workshops, including sessions for introduction, joint analysis, independent analysis, and feedback. In the introduction session, the moderator briefly introduced the goal, research background, and the SHFA method. Then, during the joint analysis, experts learned to use the method in practice by analyzing the first scenario together, with guidance from the moderator. The independent analysis starts from the second scenario onward. Experts were expected to use the method almost independently with minimal or no guidance. The moderator was available to answer questions and provide explanations if necessary. At the end, the workshops concluded with a feedback session, during which the expert participants provided their feedback on the organization, the method, suggestions, and any open questions. Feedback from a workshop was used to refine the method for the next workshop. In Workshop #4, an additional session for scenario-agnostic analysis was designed to let experts identify difficult environmental conditions merely based on the heuristic matrix without any scenario context, as a contrast experiment to the application of SHFA.

Actual Implementation and Method Refinement. Although all workshops applied the same SHFA framework with its steps, the actual execution varied slightly across each workshop. These differences can be summarized into two aspects: the scope of the implementation and the refinement of the method. Regarding the scope of implementation, in Workshops #1, #2, and #3, the moderator prepared and explained the modeled scenarios, allowing the experts to directly begin with hazardous maneuver identification (SHFA step two). In contrast, in Workshop #4, experts were guided to learn and independently perform all SHFA steps, starting from scenario modeling (SHFA step one). The primary goal of this change was to assess the

applicability of all steps of the method. The SHFA method was refined based on the issues identified during the workshops. One key refinement addressed hazardous maneuver identification (SHFA step two). We observed a lack of structure when experts derived hazardous maneuvers from intended maneuvers using guide words. To address this, we developed a look-up table to explicitly interpret the guide words and define their possible combinations with maneuvers. Another refinement focused on difficult environmental condition identification (SHFA step three). We created a heuristic matrix for each hazardous maneuver and adopted a more detailed function decomposition compared to earlier workshops, increasing the likelihood of identifying more specific difficult environmental conditions. Experts were instructed to follow the column order in the matrix, which helped structure the identification process more rigorously and reduce the chances of overlooking potential difficult environmental conditions. The details of these changes are summarized in Table 6.2.

Table 6.2.: Method Refinements between Workshops: difficult environmental condition (DEC)

<table>
<tr><th rowspan="2">WS</th><th colspan="3">SHFA step</th></tr>
<tr><th>S1: Scenario Modelling</th><th>S2: Hazardous Maneuver Ident.</th><th>S3: DEC Ident.</th></tr>
<tr><td>1</td><td rowspan="3">The scenario was modelled in advance for experts, including scenario element assortment and ego intended maneuver identification</td><td rowspan="2">Experts combined intended maneuvers and guide words to directly derive hazardous maneuvers based on their own interpretations</td><td rowspan="3">Experts identified difficult environmental conditions for each hazardous behavior with a link to the general three-layer function architecture: sense, plan, or act</td></tr>
<tr><td>2</td></tr>
<tr><td>3</td><td>Possible combinations of guide words and intended maneuver were provided, the experts selected hazardous combinations</td></tr>
<tr><td>4</td><td>Experts decomposed ego intended maneuvers based on the provided maneuver taxonomy [45]</td><td>Experts used a look-up table to derive unintended maneuvers and then select hazardous maneuvers</td><td>Experts filled difficult environmental conditions in a prepared heuristic matrix for each hazardous behavior, considering five function levels [5]</td></tr>
</table>

Due to the iterative refinement of the SHFA method, the level of detail and the application rules increased from one workshop to the next. As a result, the actual time spent on the sessions deviated somewhat from the initial plan. In particular, in workshop #4, the method was developed to be the most informative and the final version, thereby the experts were also required to perform more sub-procedures independently. As a result, the time required for learning the method during the joint scenario analysis increased significantly, surpassing the scheduled time. The time planned for each agenda point is listed in Table 6.3.

Table 6.3.: Planned Workshop Agenda

WS-ID	Introduction	Joint Analysis	Independent Analysis	Scenario-agnostic Analysis	Feedback
1 - 3	15 min	30 min	95 min	-	10 min
4	15 min	40 min	60 min	20 min	15 min

6.2. Evaluations and Findings

Feedback and suggestions from experts were collected at the end of each workshop to qualitatively evaluate the method. Besides, the identified difficult environmental conditions are analyzed for uncovering their possible clustering and revealing their distributions. These two artifacts from the workshop are discussed in the following sections.

6.2.1. Expert Feedback

Expert feedback is collected at the end of each workshop. Some feedback was directly considered and implemented in the iterative refinement of the SHFA method. The other feedback points are listed in Table 6.4. Here, we generally divide the feedback into positive aspects and open points for future working directions of the SHFA method:

Table 6.4.: Expert Feedback

ID	WS-ID	Expert-ID	Feedback
1	1	3	*"Systematic process makes it easy to follow the same routine for every scenario"*
2	1	3	*"Very valuable to do this kind of brainstorming for corner case scenarios"*
3	1	3	*"Sometimes difficult to decide when we are "done" with a scenario"*
4	2	6	*"Hazardous behavior mapping is difficult for some very fundamental/upstream functions, like localization"*
5	3	7	*"More concrete function information is needed to make the identification of difficult environmental condition easier"*
6	4	11	*"Method was clear and easy to apply"*
7	4	11	*"Since the "severity" of some trigger conditions is dependent on the system design, some assumptions on the system to look at may be helpful"*
8	4	11	*"Since we do a bottom-up approach to infer trigger conditions, I fear that a lot of effort is required to do this for many sequences. We already saw that many trigger conditions are shared by many scenarios (which all have to be evaluated for a systematic approach), I wonder how good the "efficiency" in terms of the effort to list everything to the number of final 'concluded' list of difficult environmental conditions is"*
9	4	12	*"The workshop material was very well-prepared, and the moderator went through the tasks very consistently"*
10	4	12	*"The method appears very complicated for analyzing a simple situation." Big overhead until difficult environmental conditions are really identified"*
11	4	12	*"I liked the heuristic matrix for the open brainstorming"*
12	4	12	*"Preferably, I would have liked to start with the brainstorming from the end and then go on to narrow down the thinking with the provided framework"*
13	4	13	*"Goal is clear, maybe a separate meeting for preparation in order to accelerate the work"*
14	4	13	*"More time and concentration are required for the identification of the difficult environmental conditions"*
15	4	14	*"Guidance in a separate preparation meeting?"*
16	4	15	*"With my current background, the introduction was good/enough. For even increasing efficiency, I would suggest doing some kind of introduction meeting (about 30 minutes) in order to explain everything"*
17	4	15	*"Method is systematic in every step"*
18	4	16	*"This is a good method to also include and identify the behavior-related difficult environmental conditions"*

Positive aspects. Despite the limited workshop duration (2.5 h) and the fact that most participants had no prior knowledge of the SHFA method, all experts were able to understand the method with brief guidance during the workshops and apply it to identify difficult environmental conditions from the given scenarios. Specifically, some experts explicitly mentioned that the method was clearly introduced, easy to apply, systematic, and repeatable for analyzing diverse scenarios [Feedback 1, 6, 17]. The workshop material and guidance were considered helpful for the process [Feedback 9]. The heuristic matrix used in the step of eliciting difficult environmental conditions was seen as constructive for brainstorming [Feedback 11]. Additionally, some experts acknowledged that the method can effectively support the identification of behavior-related difficult environmental conditions [Feedback 18], as it requires experts to consider whether the specific behavior of each scenario element could trigger the initial functional insufficiency during the exploration process. Furthermore, the method was regarded as a potential approach to identify corner cases [Feedback 2].

Open points and future directions. Expert feedback also indicated that the application efficiency during the workshops was limited. On one hand, a significant amount of prior knowledge (e.g., ADS safety-related terms and the scenario and function layer models) is required to conduct the method, and the method itself introduces several new concepts (e.g., maneuver transitions and the look-up table). On the other hand, the workshop schedule of only 2.5 hours for both learning and applying the method was very limited. As a result, several experts recommended conducting a separate training session beforehand to properly introduce the method and the necessary background knowledge [Feedback 12, 13, 14, 15]. We agree with this recommendation and suggest that Step #3, Eliciting Difficult Environmental Conditions, should primarily be conducted by function developers, while the preparatory procedures should be handled by safety engineers.

Some experts also noted that the scenario analysis procedures before the final identification of difficult environmental conditions created significant overhead, while difficult environmental conditions could be repeatedly identified from different scenarios, leading to low identification efficiency [Feedback 8, 10]. Our interpretation of this feedback is twofold: First, difficult environmental conditions originate from scenarios, and their identification requires the scenario context. The method aims to structure this scenario context analysis and make it traceable, which inevitably creates the necessary overhead. However, due to the very limited time and the experts' understanding of the workshop's goal to identify difficult environmental conditions, the overhead is perceived as greater than it truly is. Second, we believe it is unavoidable that different scenarios may share the same potential difficult environmental conditions. Thus, repeated efforts are not fundamentally avoidable within the method. Still, certain procedures in the SHFA method could be automated to accelerate the analysis (e.g., when the scenarios are presented as driving data, maneuver recognition can be automatically extracted using classification algorithms, as seen in [45] [77]). Furthermore, the repetition of specific difficult environmental conditions can be treated as an exposure index and used to prioritize them for testing.

During the workshops, we used the general functional decomposition model [5] instead of a concrete ADS as a reference for the final elicitation step to prevent limiting experts to analyzing overly specific functions. However, some experts reported difficulties in mapping hazardous

Type	EF1	ATTR1	BHV	INTR	EF2	ATTR2	TSC	SL	Funct. Insuf.
[Scenario#7] Quote: *Back of truck is not covered*									
I	Truck	leading, w/ open rear						4	object detection
[Scenario#9] Quote: *ego vehicle encounters other vehicles from the negated bus lane unexpectedly*									
II	Van		stops				on negated bus lane	4	HD map, behav. prediction
[Scenario#9] Quote: *oncoming vehicles that overtake*									
III	Vehicle	oncoming		overtakes	vehicle	standing	via ego lane	4	behav. prediction

Figure 6.1.: Workshop scenarios & exemplary difficult environmental conditions standardized based on original records by experts (EF: environment factor, ATTR: attribute, BHV: behavior, INTR: interaction, TSC: temporospatial context, SL: scenario layer) (©2024 Springer Nature)

behaviors to functions and identifying difficult environmental conditions with such a generic system model [Feedback 4, 5, 7]. Therefore, when using the SHFA method to identify difficult environmental conditions for a specific ADS, where the system's specification, implementation, and architecture are available, it is recommended to use concrete functions for the analysis.

6.2.2. Findings

As a result of the four workshops, 122 difficult environmental conditions were identified, and the full catalog can be found in Appendix A. These identifications were first formalized according to the methodology proposed in Chapter 4. An exemplary collection of formalized results is presented in Figure 6.1.

Clustering. Out of the 122 difficult environmental conditions identified, only 15 ($\sim 12\%$) are shared across multiple scenarios. This indicates that most of the difficult environmental conditions identified using SHFA are scenario-specific, highlighting the effectiveness of the systematic decomposition and analysis of the scenarios. However, this also suggests that not every scenario leads to unique difficult environmental conditions, implying the possibility of a saturation effect. In fact, saturation was observed at a more abstract level, as we were able to group the large number of identified difficult environmental conditions into just eight clusters that recur across scenarios (see Figure 6.2b):

#1 Ambiguity of traffic guidance , e.g., *Yellow lane marking remnant*

#2 Road user behavior , e.g., *Front truck deploys loading dock on the road*

#3 Feature/status of object , e.g., *Parking vehicle on roadside with open door*

#4 Road condition , e.g., *Slippery road*

#5 Fixed roadside structure , e.g., *Bare trees in autumn*

#6 V2X Connectivity , e.g., *Unstable mobile network*

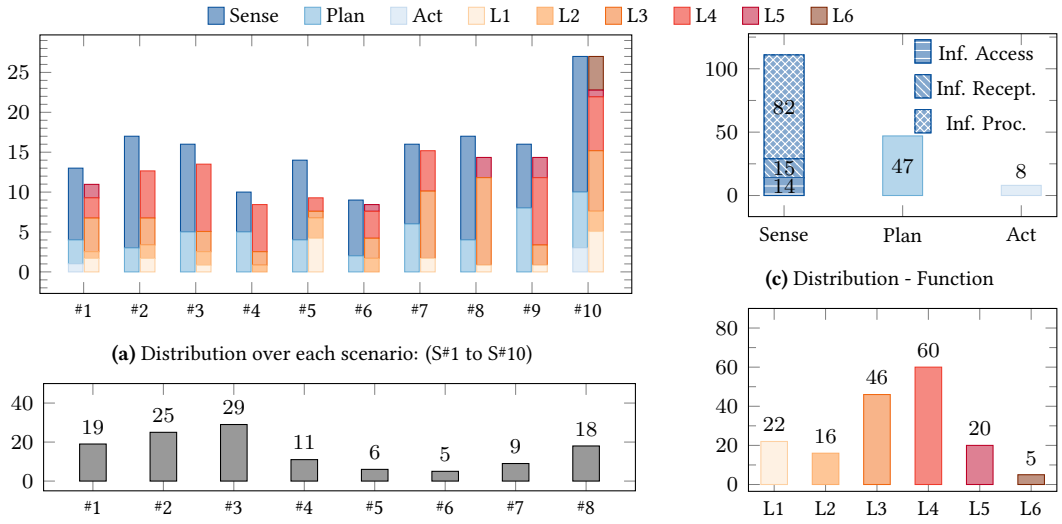
#7 Atmosphere , e.g., *Vehicle emission*

#8 Occlusion , e.g., *Roadwork barrier partially occludes crossing pedestrian*

Identifying similar aspects among different difficult environmental conditions and clustering them accordingly helps to gain a better understanding of the open world. This approach can further support the elicitation of additional difficult environmental conditions related to the clusters. On the other hand, to uncover the saturation effects at the individual level, namely showing the convergence trend in identifying new concrete difficult environmental conditions, a larger-scale analysis involving more scenarios is necessary.

Relation to system components and scenario layers. Due to procedures in the SHFA method, the identified difficult environmental conditions are naturally related to scenario layers and function components. E.g., *Front vehicle halts on the lane with hazard flasher on* is identified from the dynamic object layer [97], as experts confirmed that it might cause incorrect situational understanding by the planning components. We use these relations to compute and visualize distributions of identified difficult environmental conditions among the ten scenarios. As shown in Figure 6.2a, the majority of difficult environmental conditions are related to sensing components. Only the analysis of Scenario #4 and #9 results in equally many difficult environmental conditions for sense and plan. Most difficult environmental conditions pertain to scenario layer 3 (temporary modifications of road and of traffic guidance objects) and 4 (dynamic objects and behaviors). The least discovered is scenario layer 6 (digital information).

The distributions of difficult environmental conditions for each scenario (cf. Figure 6.2a) and the overall distribution (cf. Figure 6.2c and Figure 6.2d) reveal several findings: On the one hand, the amount of difficult environmental conditions decreases along the sense-plan-act chain. A possible reason is that functions are composed sequentially. Unintended behavior, e.g., in the planning component, can be caused by functional insufficiencies in the sensing components. This impedes identifying difficult environmental conditions for actuators, assuming that a correct action command was generated by the planning component. To investigate whether there really exist much more difficult environmental conditions related to certain components, a larger scale of experiments is needed. On the other hand, the most difficult environmental conditions pertain to dynamic objects (L4) and temporary changes of the infrastructure (L3). This could be due to the background of experts or indicate that the phenomena in these layers make automated driving difficult, as they require a robust perception and common sense reasoning beyond current traffic rules.



(a) Distribution over each scenario: (S#1 to S#10)

(b) Distribution over each cluster: (C#1 to C#8)

(c) Distribution - Function

(d) Distribution - Scenario layers

Figure 6.2.: Distribution of identified difficult environmental conditions (©2024 Springer Nature)

7. Data-driven Reconstruction of Disengagement Scenarios[1]

Knowledge-driven methods, such as the SHFA approach introduced in the previous chapter, play a vital role in comprehensively identifying difficult environmental conditions and corresponding critical scenarios. However, such methods inherently rely on experts and may be influenced by individual perspectives or experience. To compensate for the potential human bias, this chapter introduces a data-driven approach aimed at identifying critical scenarios based on real-world operational data. Specifically, an automated pipeline has been developed to reconstruct scenarios from automated vehicle disengagement events. This pipeline processes perception-generated object lists from test drives and transforms them into parameterized and directly testable scenarios. The remainder of this chapter details the concept, design, and implementation of the pipeline, highlighting its contribution to enhancing the completeness and objectivity of scenario-based safety verification.

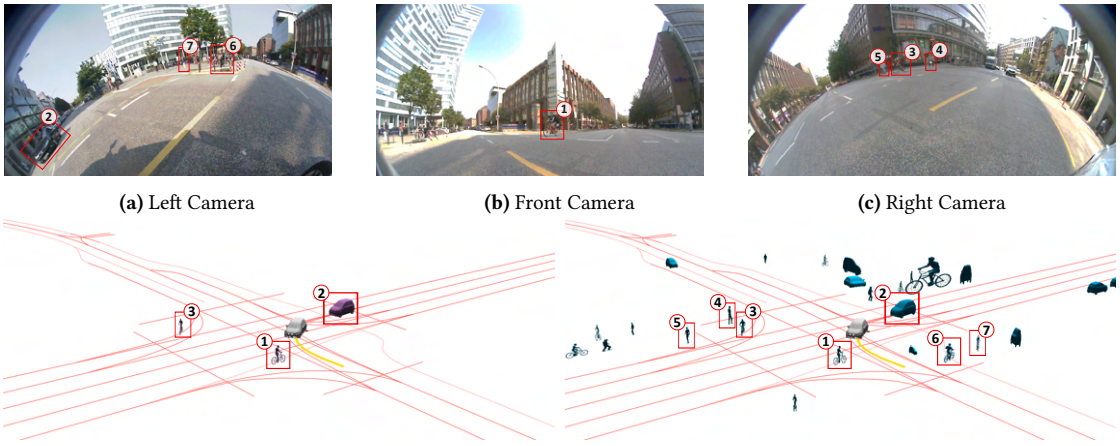
7.1. Challenges and Case Study

In 2019, the Volkswagen group deployed a fleet of automated vehicle prototypes in the city of Hamburg to conduct real-world road tests. The large number of driving hours have also captured the disengagement scenarios, where the automated driving system has failed to properly react to the traffic and the safety driver had to take over. Numerous urban driving data with environment perception measurements are collected during this campaign, serving as a foundation for a data-driven scenario elicitation.

However, there exist two main challenges to directly extract scenarios from perception measurements. On the one hand, the original measurement data represents an environment model reproduced by the perception component, which can deviate from reality due to various perception errors. Simply copying the content in the data without handling these errors can lead to unintended scenarios. On the other hand, the data usually contain too much irrelevant information, such as objects detected from a great distance without any interaction with the ego vehicle. Including too many irrelevant objects in the test scenarios firstly results in unnecessary data preprocessing effort for handling the perception errors. Secondly, the correspondingly generated scenarios require more computation power and time for the simulation. Also, it increases the difficulty in debugging the driving functions for investigating the causes of the disengagements.

[1] This chapter is based on Paper II [134] and therefore contains verbatim content previously published (©2023 IEEE).

To enable an intuitive understanding of the problems in the data and the goal of the scenario reconstruction, a disengagement scenario is illustrated as a case study. The disengagement scenario happened at an urban junction area with four ways. The ego vehicle approached the junction and intended to perform an unprotected left turn when the relevant traffic light was green. While the ego vehicle was turning left, approaching a crosswalk, a cyclist was crossing the crosswalk, and another vehicle was following the ego vehicle. Principally, the ego vehicle should have decelerated, yielded for the cyclist, then continued the left turn maneuver. However, the system failed to perform the intended deceleration and yielding in spite of the correct detection of the cyclist. So the safety driver manually disengaged the automated driving system to take over the control. One snapshot of a disengagement snippet is selected to demonstrate



(a) Left Camera **(b)** Front Camera **(c)** Right Camera

(d) Intended Scenario with 3 Objects **(e)** Original measurements with 163 Objects

Figure 7.1.: Visualization of original and revised objects in the disengagement scenario unprotected left turn (©2023 IEEE)

the scenario as can be seen from Figure 7.1. Although many other traffic participants can be observed from the three views of camera images in Figure 7.1a, 7.1b, 7.1c, the objects that are relevant for the driving tasks of the ego vehicle according to the German traffic rules are only the *Cyclist 1* crossing the crosswalk on the left and the *Cyclist 3* crossing from the right and later merging into the same lane of the ego vehicle. Additionally, the follower *Vehicle 2* should be deemed relevant for the testing, as it is a necessary condition for a rear collision. Ideally, the reconstructed scenario should only include these three objects with their correct features and trajectories as exhibited in Figure 7.1d. However, as can be seen from the visualization in Figure 7.1e, other irrelevant objects are also part of the detections.

Besides, in the close range of the ego vehicle, objects with implausible features (e.g., a huge cyclist on the upper right area), false positives (e.g., the unrealistically tiny car), and false negatives (e.g., missing pedestrians on the sidewalk on the left of the ego vehicle) are observable. Although the relevant objects were not prone to such errors in this exemplar scenario, these errors are commonly found in perception measurements and can be relevant for other scenarios. For the convenience of the discussion later, we categorize and define the major perception errors as follows:

1. **True positive inaccuracies** refer to the erroneously perceived attributes of existing surrounding objects.
2. **False positives** are the perceived objects without any reference to reality.
3. **False negatives** refer to completely or partially not perceived surrounding objects.

7.2. Scenario Reconstruction Pipeline

Based on the problems in the data as explained in the previous section, an automatic scenario reconstruction pipeline is designed as shown in Figure 7.2. The input of the pipeline is the original object list of each disengagement snippet, and the output is an OpenSCENARIO description of the reconstructed scenario.

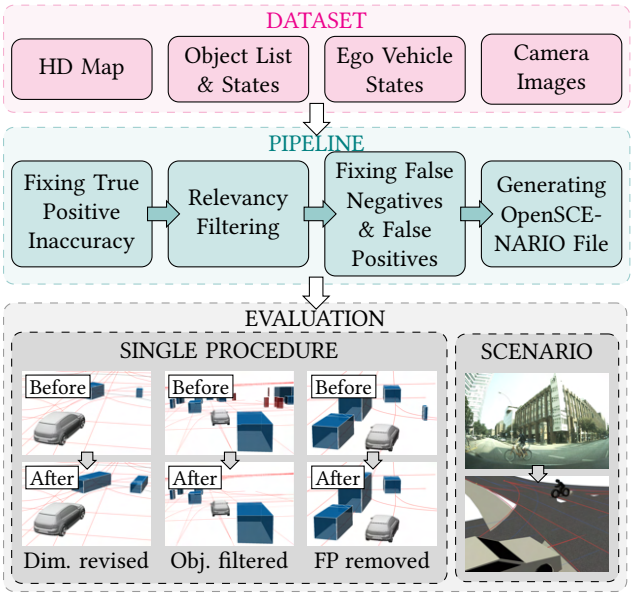


Figure 7.2.: An overview of data-based scenario reconstruction (©2023 IEEE)

The pipeline at first restores a subset of movable objects from the original object list generated by the online perception component. The restoration consists of three phases: firstly, the true positive inaccuracies are systematically handled for all objects of the input object list. Secondly, the objects deemed as irrelevant for the driving task of the ego vehicle are filtered out. Subsequently, in the scope of remaining relevant objects, false positives and objects subject to false negatives are detected and accordingly taken care of. At that point, a revised object list with all relevant objects and their plausible attributes is available. Afterwards, the pipeline generates a virtual scenario description, i.e., an OSC file, following the OpenSCENARIO standard based on the revised object list in conjunction with an OpenDRIVE file. Consequently, the OSC file comprises the inherent features and the trajectories of all scenario-relevant objects and the corresponding road network. The original ego vehicle is not part of the OSC file. To

automize the generation of OpenSCENARIO files, we utilize the open-source Python package *scenariogeneration* [6]. When it comes to e.g., regression testing with the generated scenario set-ups, a virtual system-under-test should be inserted to process the initial pose and velocity of the ego vehicle.

In the following, the three phases of dynamic objects restoration are introduced in detail.

7.2.1. True Positive Inaccuracy Revision

True positive inaccuracies can be observed in measurements of dynamic attributes like object pose and inherent attributes like classification. While the anomalies and glitches in continuous kinematic attributes can be relatively easily handled by smoothing techniques (e.g., Savitzky-Golay filter [**?**]), the falsified inherent attributes including object classifications and dimension parameters length and width are more critical, since they can significantly influence the prediction regarding object behaviors and the estimation of drivable space by the virtual controller in the test scenarios. Usually, an online perception component generates and continuously corrects assumptions about object classifications and dimensions in real time. Therefore, in the data, one object is sometimes registered with multiple different or incorrect classification assumptions and continuously changing measurements of its length and width over time. Nevertheless, for re-simulating the scenarios presented in the data, each scenario object ought to have a set of unchanged and plausible dimension parameters and a correct classification. For generating object dimension references, the most trustworthy assumptions are identified by determining the best measurement conditions. For instance, when a vehicle drives closely in parallel to the ego vehicle without another object between them, it constitutes the optimal condition for measuring the length of the vehicle. Therefore, the length assumption from this moment is taken to calculate a length reference. Each data snapshot with the states of the ego vehicle and perceived objects at each timestamp corresponds to an individual measurement condition. The conditions are quantified based on the projection angle θ (an acute angle spanned by the object heading vector and the path between the ego position and the object position), the distance d between the target object and the ego vehicle, and the occlusion proportion by another object δ as $S_\theta(\theta)$, $S_d(d)$, and $S_\delta(\delta)$, respectively. An overall confidence score of S, considering these three aspects is calculated by a weighted sum of these three parameters:

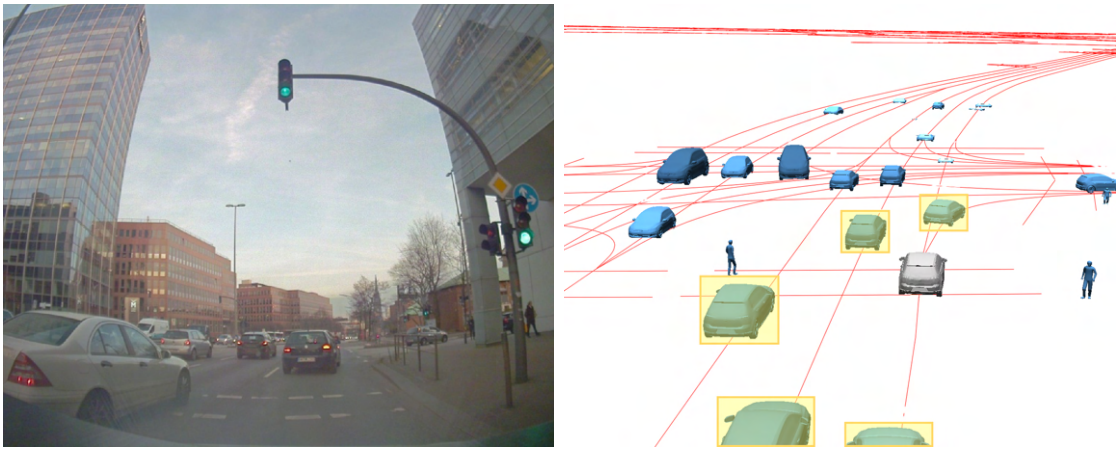
$$S = w_1 \cdot S_\theta(\theta) + w_2 \cdot S_d(d) + w_3 \cdot S_\delta(\delta) \tag{7.1}$$

$$w_1 + w_2 + w_3 = 1 \tag{7.2}$$

Finally, the conditions with the confidence scores over a pre-defined threshold are selected to calculate an average value as the dimension reference. Object classification references are deduced with a rule-based decision tree, which utilizes object velocities, calculated dimension references, and positions related to the infrastructure as classification criteria. The detailed solution is provided in our previous work [83].

7.2.2. Relevant Object Selection

The online perception component registers all detected movable objects within the detection range into the object list. In the ideal case (no occlusion, good weather, and illumination condition), the detection range is decided by the field of view of the adopted sensors. For instance, a 1550 nm band LiDAR sensor can range over 200 m [117], and thus objects could be potentially already detected at that range. However, not all objects in such a wide range are relevant for the driving task of the ego vehicle. Figure 7.3 illustrates this problem with a scenario based on a data snippet.



(a) Front Camera **(b)** Object Visualization

Figure 7.3.: Relevant and less relevant traffic participants: The leading vehicle with an activated turn indicator performs a slow *Right Turn* maneuver. The ego vehicle is required to either decelerate and wait for the leading vehicle to finish its turn maneuver or change to the left lane. The relevant objects for this scenario are highlighted. (©2023 IEEE)

For systematically selecting relevant objects, we adapt the concept of *relevant areas* proposed by Philipp et al. [82]. Firstly, the driving task of the ego vehicle in a disengagement snippet is depicted using its performed infrastructure-related maneuvers, which are identified from its kinematic measurements and an HD map with road network information. The considered infrastructure maneuvers include *Follow Lane*, *Lane Change*, *Approach Junction*, *Cross Junction*, *Turn Left*, *Turn Right*, *U-Turn*, *Approach Crosswalk* and *Cross Crosswalk*. The maneuver taxonomy and the manner of their automatic identification from measurement data are provided by Hartjen et al. [45, 44].

Then, for each identified ego vehicle maneuver, according to the position of the ego vehicle in the HD map in conjunction with traffic rules, the lanes and directions that should be paid attention to for a safe operation are determined. While this is conceptually still aligned with the work by Philipp et al. [82], the following steps differ significantly:

a. Iterating objects inside relevant areas: Philipp et al. calculate a distance threshold for each lane and direction by assuming the worst-case scenario with hypothesized objects, resulting in areas that are independent of the actual surrounding objects. For instance, a distant, slowly moving vehicle inside such a relevant area under worst-case assumptions might actu-

ally have no impact on the decisions of the ego vehicle and therefore should not be considered for the scenario reconstruction. Since we have the object list generated by a perception component, we directly iterate over the objects within the relevant lanes and then determine whether they are actually relevant based on their velocity and distance along their prospective lanes. Considering the object's current velocity and a maximum plausible acceleration according to their classification, which is determined in 7.2.1, a braking distance d_{brake} is calculated[82, 100]. Objects are considered relevant when the following condition is fulfilled:

$$d_{actual} \leqslant d_{brake} \tag{7.3}$$

d_{actual} is the distance along the lanes to either the ego-vehicle or the point of intersecting paths.

b. Considering hazard-relevant objects: Philipp et al. do not consider the direct follower of the ego vehicle as perception-relevant, since according to German traffic rules, one is not obliged to pay special attention to the direct follower. However, in scenario-based testing, the following vehicle is important because it can lead to rear collisions caused by hazardous behavior of the ego vehicle (e.g., abrupt braking), removing which can lead to not exposing some rear collision hazards. Therefore, we always regard the direct follower of the ego vehicle as scenario-relevant.

7.2.3. False Positive and False Negative Revision

The relevant object selection algorithm is neither influenced by false positives nor false negatives in the data. On the contrary, after irrelevant objects are filtered out from the object list, some false positives and some objects subject to false negatives are already removed, and therefore the revision effort is reduced. Thus, the corresponding revision is arranged after the relevancy filtering procedure.

a. Remove false positives: Since false positive objects have no reference in reality, they should also not be included in the test scenarios. Depending on the adopted perception component, both the causes and the features of false positives can be diverse. Principally, a purely data-driven method, e.g., a decision tree can be a suitable solution for deriving the features and corresponding thresholds of false positives, if the sample capacity is sufficiently high. In our case, the amount of false positives in the investigated data is not sufficient for such approaches. Therefore, a rule-based method is applied instead.

Firstly, qualitative features of typical false positives based on the adopted perception component are summarized by experts as follows:

1. Unreasonably short lifetime
2. Unreasonable velocity profile (e.g., drastic changes)
3. Active or passive crashing behavior
4. Small objects with undetermined classification

Then, we conduct a data analysis using a dataset with accumulated false positives to determine quantitative threshold values for each feature. One object is regarded as a false positive if one

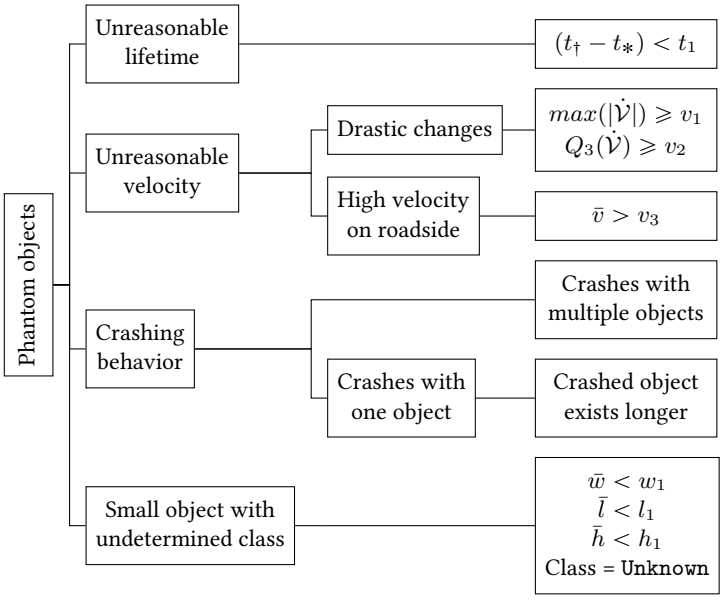


Figure 7.4.: Phantom object features (first / last timestamp t_* / $t_\dagger$, set of all velocity changes $\dot{\mathcal{V}}$, mean velocity on roadside $\bar{v}$, dimension references $\bar{w}, \bar{l}, \bar{h}$ [83]) (©2023 IEEE)

or multiple features are observed from its states. The more detailed and quantitative features for identifying phantom objects are demonstrated in Figure 7.4.

b. Revise false negatives: Some relevant objects can be completely missed or partially untracked due to adverse perception condition. Consequently, corresponding objects just appearing or disappearing suddenly in the scope of the scenario. In general, such unrealistic behavior should not be described in a virtual test scenario. Additionally, false negatives can be a long-term issue when it comes to clustering or selecting scenarios by certain criteria. For instance, if an oncoming object is detected as two temporally independent oncoming objects, while the ego vehicle is performing a left turn, the scenario should actually be categorized as *one oncoming vehicle during the ego vehicle's left turn*, but it would fall into e.g., *multiple oncoming vehicles during the ego vehicle's left turn*. Ideally, all false negatives should be taken care of. But as there is no clue to recover completely overlooked objects only based on the detection result, we limit our focus to partially missed objects.

Partially missed objects present three basic problematic fashions, namely being detected too late, disappearing too early, and disappearing for a while and appearing again with new IDs, which are named as *Late Detection*, *Early Disappearance* and *Multi-ID Problem* in the following. On the one hand, one object can be subject to one or multiple of these problems. If multiple problems are observed in one object, it is important to handle the problems in a reasonable order. On the other hand, multiple objects in one scenario can be prone to false negatives at the same time. Two independent problematic objects shall not be merged after the revision.

To tackle these problems, firstly, the relevant problematic objects and their exhibited type of false negative problems should be identified. Accordingly, a dataset with accumulated false negatives is analyzed to derive the detection criteria.

Assuming a test scenario begins at t_{sc}^{b} and ends at t_{sc}^{e}. An object O has a late detection problem, when the time of its first detection t^{b} fulfills:

$$t^{b} > t_{sc}^{b} \tag{7.4}$$

An object has an early disappearance problem when the time of its last detection t_{i}^{e} fulfills:

$$t^{e} < t_{sc}^{e} \tag{7.5}$$

If the timing t and position $\mathbf{p}$ of the early disappearance of an object O_i are close to those of the late appearance of another object O_j, and their relation fulfills:

$$0 < t_{i}^{b} - t_{j}^{e} < \epsilon \tag{7.6}$$

$$\mathrm{d}(\mathbf{p_i^e}, \mathbf{p_j^b}) < \xi \tag{7.7}$$

then O_i and O_j are identified as the same object subject to the multi-ID problem.

For objects only subject to late detection or early disappearance issues, we simply extend their trajectories based on their initial velocity v^{b} or final velocity v^{e} along the lanes where they appear or disappear accordingly. Let the length of the backwards extension for a too-late-detected object be L_{back}, then:

$$L_{back} = v^{b} \cdot (t^{b} - t_{sc}^{b}). \tag{7.8}$$

and let the length of the forwards extension for a too early disappearing object be L_{for}, then:

$$L_{forw} = v^{e} \cdot (t_{sc}^{e} - t^{e}). \tag{7.9}$$

Considering vulnerable road users on the roadside being subject to this problem, we force them to stay stationary when their trajectories and kinematic features are missing, and the existing trajectory with the corresponding object velocity $\mathbf{v_t}$ remains untouched. Namely, the object velocity should fulfill:

$$\mathbf{v(t)} = \begin{cases} 0, & t < t^{b} \\ 0, & t > t^{e} \\ \mathbf{v(t)}, & \text{otherwise.} \end{cases} , t \in [t_{sc}^{b}, t_{sc}^{e}]$$

For objects with a multi-ID problem, the IDs of the objects are aligned with the object that appears earlier. The missing trajectory between t_{i}^{e} and p_{j}^{b} is recovered by interpolating position and velocity values along the lane. Specifically, if one object is identified to have both multi-ID and early disappearance or late detection problems, the multi-ID problems should be handled first.

After the revision, the extended trajectories and corresponding kinematic features can deviate from reality. Nevertheless, such deviations are unlikely to influence the decisions of the ego vehicle in the simulation significantly, since during the time of these extensions, those objects are mostly not yet relevant for the ego vehicle.

7.3. Experiment and Evaluation

The methods explained in the previous section are applied to the experiment datasets, and the produced scenarios are used for evaluating the performance of the implemented pipeline.

Experiment set-ups. The data for the experiment originates from multiple automated test drives collected by an automated driving prototype fleet from Volkswagen Group in downtown area of Hamburg, Germany. During these test drives, the online perception components modeled the dynamic surroundings and recorded them as timestamped object lists. Each timestamp corresponds to a scene, containing a snapshot of the ego vehicle and perceived surrounding objects with their real-time attributes. Table 7.1 illustrates the attributes that are necessary for the development. Besides, videos are recorded during the test drives for investigating the perception ground truth. Additionally, an HD map is used for an overall description of the road network, the online localization of the ego vehicle, and the lane matching of detected objects. An OpenDRIVE file corresponding to the location of the test drives is also available and can be used for road network description in virtual scenario testing. With such data resource

Table 7.1.: Relevant object attributes in the adopted dataset (©2023 IEEE)

Attribute	Unit	Content
ID	-	6-digit ID of the detected object
Global easting	m	Object position in UTM coordinates
Global northing	m	Object position in UTM coordinates
Global height	m	Object position in UTM coordinates
Yaw	rad	Object heading in the ego coordinates
v_x	$\mathrm{m\,s^{-1}}$	x-component of obj. velocity in the ego coordinates
v_y	$\mathrm{m\,s^{-1}}$	y-component of obj. velocity in the ego coordinates
Lane ID	-	Object position matched lane ID in the HD map
Length	m	Object length
Width	m	Object width
Height	m	Object height
Class	-	Object classification

availability, we focus on reconstructing the dynamic part of test scenarios based on the object lists and the HD map, while the static road network is introduced by directly importing the OpenDRIVE file.

As we aim at reconstructing disengagement scenarios, the disengagement moments are identified from the test drives by detecting the switching of driving modes. Then, disengagement snippets are obtained by extending 5 s backwards and forwards from the disengagement moments. In total, 146 disengagement snippets are collected. Among all, 137 disengagement snippets are utilized for developing and implementing the concepts, while the rest 10 snippets are used for validation. Because the amount of false positives and false negatives in the disengagement snippets is limited, two additional sets of driving snippets without disengagements but containing more false positives and negatives are prepared for development purposes. In summary, the experiment datasets comprise 3 development datasets $\mathcal{D}_{sc}$, $\mathcal{D}_{fp}$, and $\mathcal{D}_{fn}$ and one validation dataset $\mathcal{V}_{sc}$. An overview of these datasets is provided in Table 7.2.

Table 7.2.: Overview of development and validation datasets (©2023 IEEE)

Dataset	Content
$\mathcal{D}_{sc}$	137 disengagement snippets, 10 s each
$\mathcal{D}_{fp}$	accumulative data snippets with 142 FP objects
$\mathcal{D}_{fn}$	accumulative data snippets with 194 FN objects
$\mathcal{V}_{sc}$	10 disengagement snippets, 10 s each

Evaluation. For qualitatively and intuitively illustrating the performance of the scenario reconstruction pipeline, we reuse the disengagement snippet from Section 8.3. These perception errors and irrelevant objects are cleaned up, which result in more plausible features and a reduced number of objects in the scene. An OpenSCENARIO file is generated to describe the revised object list and is imported into a simulator called esmini [64] to visualize the virtual test scenario in Figure 7.5. It can be seen that the simulation is aligned with the revised scenario except for the ego vehicle, since no controller is placed into the test scenario yet. Apart from the case study, 4 more representative disengagements and their corresponding simulated scenarios are presented in Table 7.4.

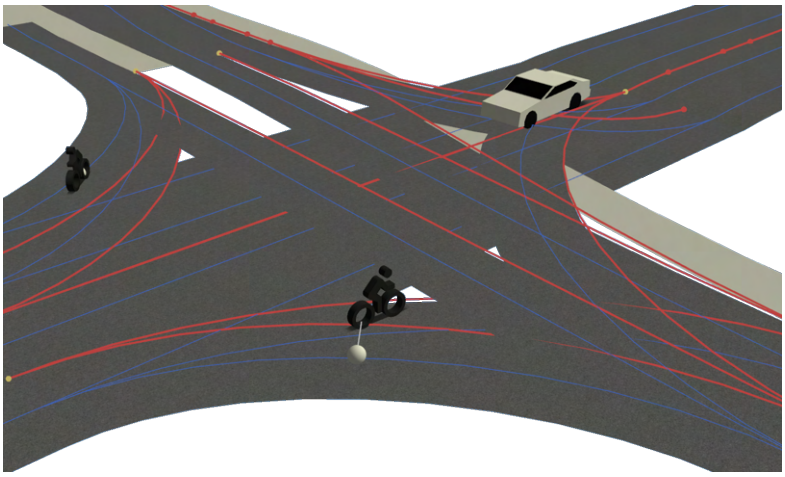


Figure 7.5.: Simulation of our case study scenario with esmini (©2023 IEEE)

Since it is challenging to quantitatively assess the overall performance of the pipeline, we analyze the performance of the sub-functions within the pipeline instead to indirectly reflect the pipeline's correct functioning. Therefore, the automatically selected and revised objects in the reconstructed scenarios are compared with the manually annotated ground truth labels for scenario-relevant objects, false positives, and objects subject to false negatives in the validation dataset $\mathcal{V}_{sc}$. Referring to the concept of *completeness* by Topan et al. [109], the perception result should at least include all safety-critical objects. We adopt the same criterion for selecting relevant objects. Therefore, it is intended that at least all annotated relevant objects are selected. Regarding false positives, the criterion *soundness* [109] is targeted, namely, all removed objects should correspond to annotated false positives. Besides, for false negatives, we validate the successful extension of the trajectories of all objects with early disappearance and late detection

problem and ensure that objects with multi-ID problems are revised to have consistent IDs without forcing independent objects into one object. Table 7.3 provides the results of relevant object selection, false positive revision and false negative revision against the corresponding number of ground truth labels as well as the number of objects in the scenarios before and after the pipeline processing.

Table 7.3.: Quantitative evaluation based on $\mathcal{V}_{sc}$ (©2023 IEEE)

#	Relevant Objects		False Positives		False Negatives		Obj. No.	
	Precision	Recall	Precision	Recall	Precision	Recall	Orig.	Rev.
1	3 / 6	3 / 3	1 / 1	1 / 1	1 / 1	1 / 1	123	4
2	4 / 5	4 / 4	-	-	-	-	39	5
3	2 / 7	2 / 2	1 / 1	1 / 1	-	-	108	6
4	9 / 14	9 / 9	3 / 3	3 / 3	-	-	105	11
5	10 / 12	10 / 10	1 / 1	1 / 1	-	-	126	11
6	3 / 13	3 / 3	5 / 5	5 / 5	-	-	137	8
7	3 / 7	3 / 3	1 / 1	1 / 1	-	-	65	6
8	2 / 7	2 / 2	2 / 2	2 / 2	-	-	140	5
9	10 / 17	10 / 10	6 / 6	6 / 6	-	-	148	11
10	4 / 6	4 / 4	2 / 2	2 / 2	-	-	163	4

7.4. Limitations and Discussions

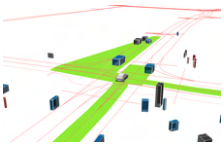
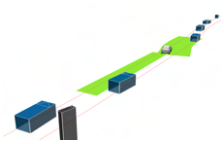
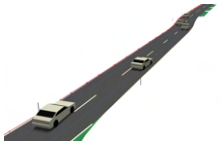
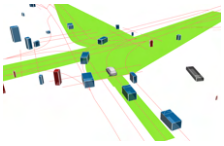
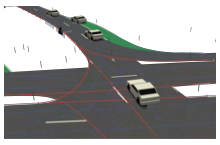
According to the previous section, the automated pipeline exhibits sufficient performance for filtering irrelevant objects and repairing several major perception errors in the remaining relevant objects. However, limitations regarding the following aspects should be put into notice:

Handling clueless perception errors. To identify phantom objects or correct wrong attributes of existing objects in the object list, sufficient evidence and some correct measurements need to be available in the data. If an existing object is not detected or is always perceived to have wrong measurements, there is no way to fix the error merely based on the object list. Therefore, if such objects cause the disengagement, the pipeline cannot generate the correct critical scenarios to reveal the reality. For example, the completely undetected objects are not a part of the output scenario, which makes the scenario different from the original disengagement scenario and should not be used in the regression test.

Other possible reasons for disengagement. Disengagements could be triggered by functional insufficiencies and caused by electrical and electronic failures of the driving system, crashing software or hardware, or even wrong operations by safety drivers (e.g., too early or unnecessary intervention). Since the pipeline is only designed to support the investigation of functional insufficiencies, disengagements caused by other factors are not the target data for the pipeline. A preliminary disengagement selection is the precondition for using the pipeline.

Connection to difficult environmental conditions. The pipeline is designed to extract critical scenarios, which could contain difficult environmental conditions. However, identifying the causes of the disengagements or the relevant difficult environmental conditions relies on a manual debugging process by function developers with full access to the system design.

Table 7.4.: Four exemplary disengagements (Camera images and objects come from the online perception, while green relevant areas are calculated offline to further identify relevant objects and finally extract the underlying scenario.) (©2023 IEEE)

	Camera Image	Scene Visualization	Extracted Scenario	Comment
(a) Left Turn				This scenario requires the ego vehicle to perform an unprotected left turn at a 4-way intersection with two oncoming vehicles. In parallel, another vehicle is passing on the right. The extracted scenario comprises these relevant traffic participants, while most other objects are filtered out.
(b) Lane Change				This scenario requires the ego vehicle to perform a lane change to the left because of a standing vehicle in front. At the same time, there is an incoming vehicle from behind so that the ego vehicle's velocity shall not significantly decrease. Both these vehicles are considered in the scenario.
(c) Right Turn				This scenario requires the ego vehicle to perform a right turn. It is surrounded by a motorbike, following vehicles and oncoming vehicles. Moreover, a pedestrian is standing close to the road. These objects are featured within the scenario, while other objects and false positives are filtered out.
(d) Follow Lane				This scenario requires the ego vehicle to keep its lane. However, a parallel vehicle is close to intruding to the ego vehicle's lane because of a wide van standing at the side of the road. These two vehicles as well as a parked bike and an approaching vehicle from behind are part of the scenario.

Still, the automated pipeline can accelerate the overall investigation of the disengagements by re-simulating the extracted scenarios: When the extracted scenarios lead to the hazardous behavior avoided by the disengagement, the difficult environmental conditions are related to the constellated situation by road users. The debugging should direct the planning components. When the hazardous behavior does not occur in the extracted scenarios, the difficult environmental conditions are related to particular objects and filtered out by the pipeline. Thus, the investigation should focus on the difference between the original and the extracted scenarios, and direct the perception components. Nevertheless, the re-simulation for investigating the disengagements and the related system is not possible, as the relevant software stack used in the field experiments for this case study was not available at the time of the pipeline development.

Part IV.

Testing and Evaluation of Difficult Environmental Conditions

8. Injection of Difficult Environmental Conditions into Scenarios[1]

Difficult environmental conditions are identified and managed as independent working artifacts. Ultimately, their purpose is to test a target automated driving system and expose potential functional insufficiencies. A difficult environmental condition can be applicable across various scenario contexts. Conversely, a specific scenario may also include diverse difficult environmental conditions. Additionally, there are numerous possibilities for parameterizing a difficult environmental condition. For example, one could vary parameters like velocity, trajectory, or even dimensions (e.g., height) when considering *A jaywalker*. This chapter presents a systematic approach for integrating difficult environmental conditions into test scenarios, along with their parameterization.

8.1. From Difficult Environmental Conditions to Scenarios

As introduced in Section 2.5, one can either extend a difficult environmental condition to a scenario by selecting some ODD elements and adding on corresponding actions of objects (an extension-based approach) or combine a difficult environmental condition with tested, hazard-free scenarios (a combination-based approach).

While the extension-based approach is intuitive, it has a significant drawback: by attaching scenario contexts (including ODD elements and their actions) to a target difficult environmental condition, there is a risk of introducing unknown environmental conditions. If the system fails the test, it becomes unclear whether the failure is due to the target difficult environmental condition or the extended context. For instance, to test the perception robustness of ADS in rainy weather, a scenario is created where the ego vehicle follows a cyclist in the rain. If the system fails to maintain a safe distance, it is uncertain whether the issue is caused by the rain or the system's inability to safely respond to cyclists in general.

The combination-based approach can mitigate this problem, which pairs difficult environmental conditions with scenarios the ADS have already passed. For example, if the ADS successfully handles the scenario of the ego vehicle following a cyclist in normal conditions, that scenario can be used as a reference. Suppose the ADS fail the same scenario under rainy conditions. In that case, it indicates that the system is not robust to rain, confirming that rainy weather is a valid difficult environmental condition and suggesting the need for functional modifications in the perception components.

[1] This chapter is based on Paper IV [131] and therefore contains verbatim content previously published (©2025 SAE).

From an industrial development perspective, the combination-based approach is more advantageous where time, cost, traceability, and efficiency are critical. It allows for parallel development of scenario and difficult environmental condition catalogs, whereas the extension-based approach requires sequential development. In the latter, scenario development would be dependent on the progress of difficult environmental conditions, potentially slowing down the overall process. For these reasons, the combination-based approach is preferred in this thesis.

8.2. Generation of Potential Critical Scenarios

Following the combination-based approach, difficult environmental conditions are combined with a tested, hazard-free reference scenario to derive potential critical scenarios. In this section, we introduce a systematic method to guide the selection of suitable scenarios for the combination process and the subsequent scenario generation.

8.2.1. Abstraction Levels and Procedures

As elaborated in Chapter 3, difficult environmental conditions are specific conditions of scenarios. Thus, the three abstraction levels (functional, logical, and concrete [74]) for developing scenarios also apply for describing and parameterizing difficult environmental conditions. Our scenario generation method starts from the functional level (cf. Figure 8.1), as depicted in the following:

Functional level: check compatibility. Before the combination, a difficult environmental condition and a scenario should be assured compatible to each other. To do so, we specify that (1) a certain context required by the difficult environmental condition should be available in the scenario to avoid a fundamental change of the scenario (defined as *Compatibility Check*). An example of violating (1) would be a difficult environmental condition *Front vehicle in the neighbor lane drives on lane markings*, given a scenario where the road consists of a single lane. Adding the difficult environmental condition requires adding a new lane and thus changes the scenario completely. To further reduce the variance of the scenario, we additionally examine whether (2) the road users' original behaviors and trajectories can stay unchanged despite the inclusion of the difficult environmental condition (defined as *Variance Control*). Checking these two aspects can be conducted on a semantic level. As a result, only compatible pairs of potential difficult environmental conditions and scenarios will be combined to create test scenarios.

Logical level: determine relevant parameter space. Difficult environmental conditions are decomposed into scenario elements and relevant parameters according to their linguistic meaning. To fit into a reference scenario, parameter ranges of a difficult environmental condition are restricted by the parameter space of the reference scenario. E.g., if a difficult environmental condition is *a faded lane marking*, the position and size of the lane marking should refer to how they are specified in the reference scenario. On the one hand, the parameters should be specified in a way that the difficult environmental condition is relevant for the ego vehicle. Therefore, the parameterization of a difficult environmental condition is highly dependent on the kinematic parameters of the ego vehicle (e.g., its velocities and the starting position) in its corresponding reference scenario. On the other hand, to assure that the behaviors and trajec-

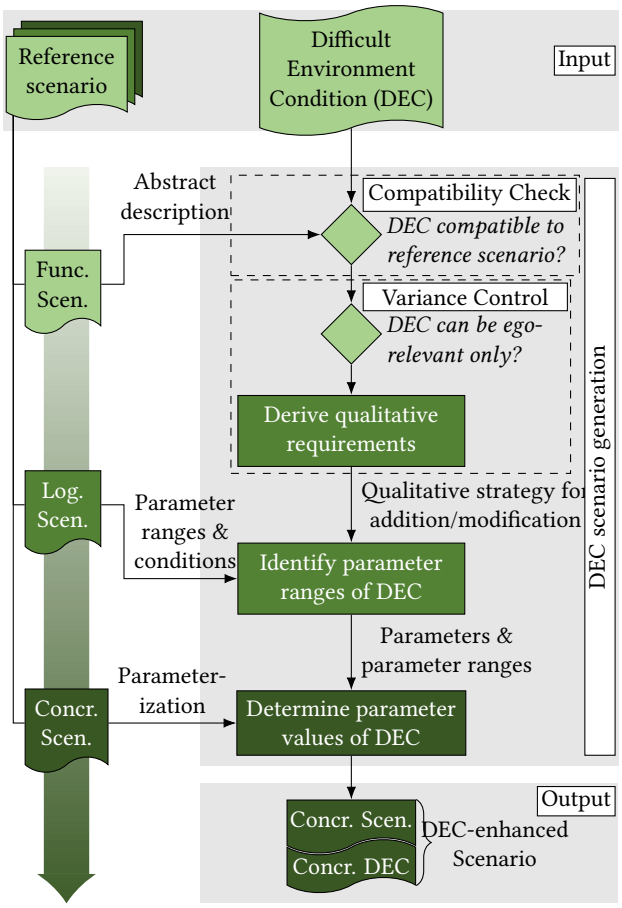


Figure 8.1.: Overview of difficult environmental condition (DEC) scenarios generation (©2025 SAE)

tories of other road users in the reference scenario do not change, further restrictions to the parameter ranges of the difficult environmental condition are considered.

Concrete level: finalize test scenarios. A concrete test scenario is generated as a variance to the concrete reference scenario. The difficult environmental condition related parameters are instantiated with new values, while the other parameters inherit the values from the reference scenario. Usually, due to limited test resources, exhaustive exploration of the whole parameter ranges is too expensive and inefficient due to possible similarities. Instead, concrete values of difficult environmental condition parameters are sampled from a previously derived parameter space according to certain criteria to create concrete test scenarios. For instance, the sampling process can be conducted according to experts' knowledge or statistical methods, and there exist already many works in these directions, e.g., [22, 103, 16]. Therefore, we focus on the white spots and introduce an approach for scenario generation on functional and logical levels in the following.

8.2.2. Compatibility Check & Variance Control

Difficult environmental conditions and reference scenarios are both related to several scenario elements. The scenario elements described in a difficult environmental condition are regarded as *DEC-relevant elements*. Scenario elements in the reference scenario that are not identical to the *DEC-relevant elements*, are deemed as *DEC-irrelevant elements*. While DEC-relevant elements should be added or modified in the reference scenario for including the difficult environmental condition, DEC-irrelevant elements should remain possibly unaffected for restricting the scenario variance. Meanwhile, scenario elements can be grouped into different scenario layers according to a model defined within the German research project PEGASUS [36] as following:

$\mathcal{L}_1$ Road model, e.g., road geometry, topology, and surface.

$\mathcal{L}_2$ Traffic infrastructure, e.g., barriers and traffic signs.

$\mathcal{L}_3$ Temporary modification of $\mathcal{L}_1$ and $\mathcal{L}_2$.

$\mathcal{L}_4$ Dynamic objects, including their static features (e.g., positions) and dynamic aspects (e.g., maneuvers).

$\mathcal{L}_5$ Environment condition, including weather, lighting, and other surrounding conditions.

Thus, a difficult environmental condition can relate to multiple scenario layers. Exemplarily, *Roadwork barrier partially occludes the crossing cyclist* is related to scenario layer $\mathcal{L}_3$ (*Roadwork*) and $\mathcal{L}_4$ (*Crossing cyclist*). The compatibility check is conducted on corresponding scenario layers with a criterion:

Criterion 1. Necessary context for including DEC-relevant elements (and their corresponding features, actions, and interactions) should be available in the reference scenario.

More specifically, post-adding certain DEC-relevant scenario elements is forbidden, as the addition of those will notably change the scenario essence. We specify the not post-addable scenario elements with expert knowledge: All elements related to road models ($\mathcal{L}_1$) and traffic guidance/regulation objects ($\mathcal{L}_2$) are not addable. Dynamic objects ($\mathcal{L}_4$) are not addable if a difficult environmental condition depicts merely a static feature of such objects (e.g., a pedestrian with unusual height). If a difficult environmental condition is about an action of a dynamic object and the action is not performed in the reference scenario, post-adding the object with the action is allowed and is preferred over modifying the action of an existing object.

Post-adding or modifying is allowed for the other scenario elements. Still, the variance of the reference scenario should be minimized following a further criterion:

Criterion 2. DEC-irrelevant elements and their original behaviors should stay plausible in the context of the new scenario.

In practice, we specify that the modifications in scenario layers $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$, and $\mathcal{L}_5$ should not result in different traffic rules or drivable areas for the *DEC-irrelevant objects* compared to those in the original reference scenario. Meanwhile, post-added roadside infrastructures ($\mathcal{L}_2$),

temporary additions ($\mathcal{L}_3$) and dynamic objects ($\mathcal{L}_4$) should not interfere with the trajectories of the *DEC-irrelevant objects*. The concrete rules based on these two criteria are defined and shown in Table 8.1.

Table 8.1.: Difficult environmental condition (DEC) combination rules for each scenario layer (©2025 SAE)

Scenario layer		Compatibility Check (CC)	Variance Control (VC)
$\mathcal{L}_1$	Road Model	The road model required by DEC should be available in the original scenario, including: **CC-1.1**: Road layout (how roads connect each other) **CC-1.2**: Road element (road segment, junction, crosswalk, etc.) **CC-1.3**: Road markings (guidance marking on road surface) **CC-1.4**: Number of lanes in a road **CC-1.5**: Direction of lanes	**VC-1** Deleting/modifying an element from the road model required by DEC should not require a different driving policy for the DEC-irrelevant objects
$\mathcal{L}_2$	Traffic Infrastructure	**CC-2** The required permanent traffic guidance objects (e.g., traffic signs/lights) by DEC should be available in the original scenario	**VC-2.1** Added roadside infrastructure or road furniture should not intersect with the trajectory of DEC-irrelevant objects **VC-2.2** Modifying traffic signs/lights required by DEC should not require a different driving policy for the DEC-irrelevant objects
$\mathcal{L}_3$	Temporary Modification of L1/L2	**CC-3** Temporary modifying/negating traffic guidance markings, signs, or traffic lights required by DEC, corresponding traffic guidance markings, signs, or traffic lights should be available in the original scenario	**VC-3** The DEC-required temporary modification is only relevant to the ego vehicle **VC-3.1** Added construction side/roadworks/obstacles required by DEC should not intersect with the trajectory of the DEC-irrelevant objects **VC-3.2** Adding temporary traffic marking/sign/light required by DEC should not require a different driving policy for DEC-irrelevant objects
$\mathcal{L}_4$	Dynamic Objects	**CC-4.1** If the DEC requires a static feature of a dynamic object, the object should already exist in the original scenario **CC-4.2** The road geometry and space required by the position or travelling direction of the DEC-relevant dynamic object should be available in the original scenario	**VC-4** Dynamic objects with specified action are required by the DEC: **VC-4.1** If the required object action is not already performed in the original scenario, a new dynamic object with the action should be added into the scenario **VC-4.2** The velocities/positions/trajectories/maneuvers of the existing dynamic objects should not change
$\mathcal{L}_5$	Environment Condition		**VC-5** The DEC should not violate the weather and time conditions in the original scenario

8.3. Case study

To demonstrate our proposed method in Section 8.1, we provide a case study in both depth and breadth: firstly, we illustrate how the method step-by-step works for one reference scenario with details. Namely, how a difficult environmental condition is selected out of the given candidates and integrated into the reference scenario for testing. Secondly, we examine the applicability of the method on a larger scale. Multiple difficult environmental conditions and scenarios are analyzed regarding their compatibility and are combined with their corresponding reference scenarios.

8.3.1. Detailed Illustration with One Reference Scenario

Experiment settings. The reference scenario depicts an urban driving situation: the ego vehicle (Vehicle #1) follows Vehicle #2, which drives in front of the ego vehicle at a short distance. The road has straight geometry and consists of Lane #1 for the traveling direction of the ego vehicle and Lane #2 for the opposite direction. The scenario happens during the daytime under clear weather conditions. Details of the scenario in five scenario layers and three abstraction layers are provided in Table 8.3. Three candidate difficult environmental conditions are:

DEC_1 *Roadwork in the middle of the lane*

DEC_2 *Fully occluded relevant right-of-way sign*

DEC_3 *Oncoming vehicle overtakes a halting vehicle*

Rule checking. Accordingly, DEC-relevant elements for DEC_1, DEC_2, and DEC_3 are a *roadwork ($\mathcal{L}_3$)*, a *right of way sign ($\mathcal{L}_2$)*, and two *oncoming vehicles with interaction ($\mathcal{L}_4$)* individually. Therefore, the rules of the corresponding scenario layers are examined against the scenario description (cf. Table 8.2). While DEC_1 and DEC_2 are directly excluded for the given

Table 8.2.: Detailed rule checking (©2025 SAE)

ID	Required Element	Compatibility Check & Variance Control	
		Rule	Result
DEC_1	Roadwork	CC-n/a	No rule against adding roadworks in general
		VC-3.1	Failed. Roadwork on Lane #1 will intersect the trajectory of Vehicle #2
DEC_2	ROW sign	CC-2	Failed. Ref. scenario contains no traffic sign
DEC_3	Oncoming vehicles with specific actions	CC-4.2	Passed. Lane for oncoming traffic available
		VC-4.1	Conditional pass: Adding oncoming vehicles with required actions
		VC-4.2	Conditional pass: Overtaking trajectory shall not intersect the trajectory of Vehicle #2

Table 8.3.: Scenario description & parameterization (©2025 SAE)

(a) Reference scenario

S. L.	Functional	Logical		Concrete*
		Param.	Range*	
$\mathcal{L}_1$	Road #1 has a straight geometry of a **short** length	L	[150,500]	308
	Road #1 has in the direction of travel Lane #1, on the opposite direction of travel Lane #2			
$\mathcal{L}_2$	No infrastructure defined			
$\mathcal{L}_3$	No temporary modification defined			
$\mathcal{L}_4$	Ego Vehicle #1 is located on Lane #1	s_e	[0,L]	$s_e(t_0) = 50$
	Ego Vehicle #1 drives with **normal** velocity	v_e	[0,13.89]	$v_e(t_0) = 5.56$
	Vehicle #2 is located on Lane #1 and drives **in front of** Vehicle #1	s_2	(s_e,L]	$s_2(t_0) = 120$
	Vehicle #2 drives with **normal** velocity	v_2	[0,13.89]	$v_2 = 8.33$
$\mathcal{L}_5$	The scenario during daytime	ToD	[6,18]	12 a.m.

(b) Difficult environmental condition DEC_3

S. L.	Functional	Logical		Concrete*
		Param.	Range*	
$\mathcal{L}_4$	Vehicle #3 drives on Lane #2 with **normal** velocity. Then it overtakes Vehicle #4 via Lane #1, when it reaches **a specific distance** behind Vehicle #4	$s_3(t_0)$	[$s_4 + d_s$,L]	to be varied
		v_3	[0, 13.89]	to be varied
		d_s	(0,L-s_4)	30
	Vehicle #4 halts on Lane #2	s_4	[$s_e(t_0)$,L)	120

∗ **in SI units if available:** L: length, s: s component of position in Frenet coordinates, v: velocity, d: distance, ToD: time of day, t_0: start time of scenario

scenario due to violating the variance control rule **VC-3.1** and the compatibility rule **CC-2**, respectively, integrating DEC_3 complies with the compatibility rule **CC-4.2** and can meet the relevant variance control rules **VC-4.1** and **VC-4.2**.

Implementation based on DEC_3. To combine the difficult environmental condition DEC_3 with the given scenario, an oncoming overtaking Vehicle #3 and a halting Vehicle #4 are added on Lane #2 (cf. Rule **VC-4.1**). Also, the scenario description on the functional level is updated. An exemplary summary is *Ego vehicle follows Vehicle #2 on Lane #1. Vehicle #3 drives, and Vehicle #4 halts on Lane #2. Subsequently, Vehicle #3 overtakes Vehicle #4 via Lane #1.* Furthermore, Vehicle #3 and its overtaking action should be relevant for the decision-making of the ego vehicle. The overtaking trajectory should not intersect with the trajectory of Vehicle #2, since Vehicle #2 is a DEC-irrelevant object (cf. Rule **VC-4.2**). Thus, two qualitative requirements are derived:

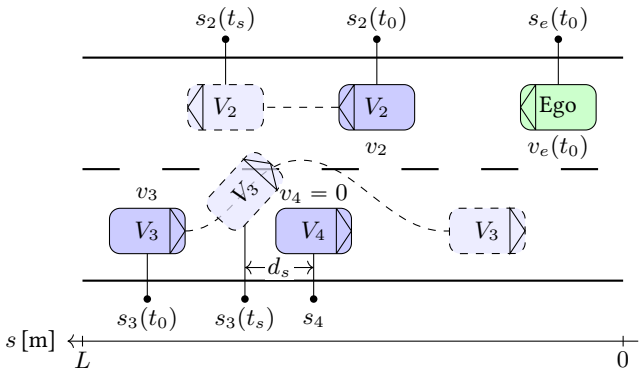


Figure 8.2.: DEC_3-enhanced scenario visualization (©2025 SAE)

Req.1 Vehicle #3 should start overtaking in front of the ego vehicle with adequate distance.

Req.2 Vehicle #2 should already pass Vehicle #3, when Vehicle #3 starts entering Lane #1.

DEC_3 relevant parameters should depict the kinematics of the two relevant vehicles (Vehicle #3 and #4), e.g., their positions $s_3(t)$, s_4 and velocities $v_3(t)$, v_4 ($v_4 = 0$) over time t. Due to the two requirements, the parameters above are also restricted by the kinematics of the ego vehicle and Vehicle #2. Additionally, secondary parameters t_s and d_s specify the timing when Vehicle #3 starts to enter Lane #1 and the longitudinal distance between Vehicle #3 and Vehicle #4 at t_s. For the sake of simplicity, we specify constant velocities v_2 and v_3 for Vehicle #2 and Vehicle #3 during the whole scenario. The scenario is visualized in Figure 8.2.
As a result, the requirement Req.1 can be formulated as:

$$s_3(t_0) > s_4 + d_s \tag{8.1}$$

$$s_4 > s_e(t_0) \tag{8.2}$$

For Req.2, at t_s the following conditions should be fulfilled:

$$s_2(t_0) + v_2 \cdot t_s > s_4 + d_s \tag{8.3}$$

$$s_3(t_0) - v_3 \cdot t_s = s_4 + d_s \tag{8.4}$$

Let $s_3(t_0)$ and v_3 be variables and others be fixed coefficients, with (8.1)-(8.4) there is:

$$\underbrace{\frac{v_2}{s_4 + d_s - s_2(t_0)}}_{\text{a}} \cdot \underbrace{s_3(t_0)}_{\text{x}} \underbrace{- \frac{v_2 \cdot (s_4 + d_s)}{s_4 + d_s - s_2(t_0)}}_{\text{b}} > \underbrace{v_3}_{\text{y}} \tag{8.5}$$

Parameters in Inequality (8.5) are restricted to a set of ranges according to the settings of the reference scenario (cf. Table 8.3a), e.g., positions of Vehicle #3 are restricted by the length L of its driving lanes. The following ranges are considered:

$$L \in [150, 500]\text{m} \tag{8.6}$$

$$v_2, v_3 \in [0, 50]\text{km h}^{-1} \tag{8.7}$$

$$s_e(t_0), s_2(t_0), s_3(t_0), s_4 \in [0, L]\text{m} \tag{8.8}$$

$$s_2(t) > s_e(t) \tag{8.9}$$

To finalize a concrete test scenario of DEC_3, the parameters L, $s_e(t_0)$, $v_e(t_0)$, $s_2(t_0)$, v_2 inherit values from the concrete reference scenario. Furthermore, we specify the values for the fixed coefficients in Inequality (8.5) (d_s and s_4) from their ranges (cf. Table 8.3b). Subsequently, the feasible area for the variable parameters $s_3(t_0)$ and v_3 is derived and illustrated in Figure 8.3.

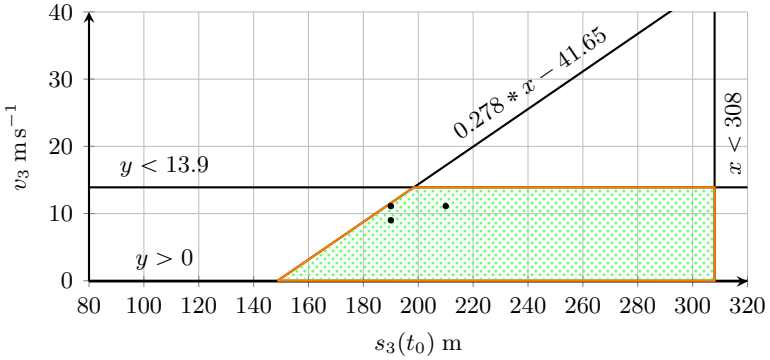


Figure 8.3.: Parameter ranges for DEC_3: Lower and upper limits for v_3 refer to Condition (8.7), limits for $s_3(t_0)$ refer to Condition (8.8). The linear constraint corresponds to Inequality (8.5). (©2025 SAE)

Result demonstration. For illustration purposes, experts manually select three sets of values for the variables in Inequality 8.5, namely the starting position $s_3(t_0)$ and the velocity v_3 of Vehicle #3, considering the effects of them theoretically. The three sets of values are marked in the defined parameter space in Figure 8.3 and are listed as follows:

$$(s_3(t_0),v_3) = \begin{cases} (190,9), & \text{Test case } \mathcal{A}. \\ (190,11.11), & \text{Test case } \mathcal{B}. \\ (210,11.11), & \text{Test case } \mathcal{C}. \end{cases} \tag{8.10}$$

The corresponding scenarios are tested with an automated driving software stack, Apollo [9], in a simulator, Carla [24]. The Apollo system takes perfect environmental information. Therefore, the test focuses on verifying the potential difficulties of the planning component. The presence or absence of collisions between the ego vehicle (in black) and any other object is defined as a pass/fail criterion. The snapshots of the tests are sampled based on several important timestamps, namely at t_0 when the scenario starts, at t_1 when Vehicle #2 (in red, which is in front of the ego vehicle) passes Vehicle #4 (in green, halting on the other lane), at t_2 (previously defined as t_s) when the Vehicle #3 (in white, overtaking) starts the lane change maneuver, at t_3 when the ego vehicle recognizes the lane change intention of Vehicle #3 and starts breaking, and at t_4 when the ego vehicle fully stops.

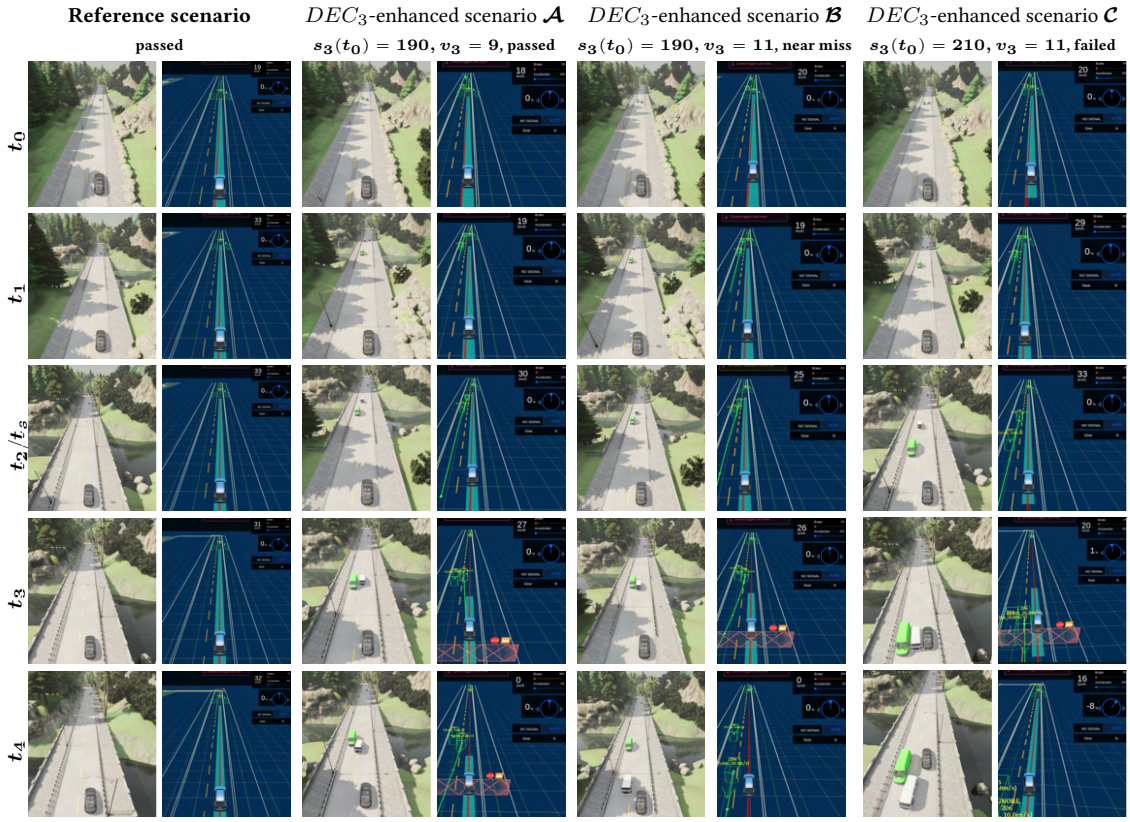


Figure 8.4.: Simulation test results of DEC_3 (©2025 SAE)

As shown in the simulation result (cf. Figure 8.4), in $\mathcal{A}$, the system detects the intention of Vehicle #3 to change lane at t_3, while Vehicle #3 is 30 m ahead. The system fully stops with 16.3 m distance to Vehicle #3 with a relatively mild deceleration profile. Thus, $\mathcal{A}$ is regarded as being passed. As a control group to $\mathcal{A}$, Vehicle #3 in $\mathcal{B}$ is assigned with increased velocity. The system starts to brake with a 25 m distance to Vehicle #3 and completely stops with only a distance of 5.6 m. The braking is more drastic and almost not in time. The system fails in $\mathcal{C}$. In comparison to $\mathcal{B}$, Vehicle #3 starts from a further away position. Consequently, the distance between the ego vehicle and Vehicle #3 since t_2 is smaller due to more travelling time before. Finally, the system fails to recognize the lane change of Vehicle #3 before a collision.
In fact, a delay in the prediction is observed in all three test cases by comparing the predicted trajectories of Vehicle #3 at t_2 and t_3. More specifically, the system will take action only when the predicted trajectory of a target vehicle intersects the defined driveable area (the light blue blocks in Figure 8.4) of the ego vehicle. This delayed prediction finally caused the failure in test scenario $\mathcal{C}$, and we analyze the related functional insufficiencies further: On the one hand, an inaccuracy in the predicted trajectory is observed. Referring to the visualization at t_2 in test $\mathcal{C}$, there is a deviation between the actual heading and the predicted trajectory of Vehicle #3. On the other hand, the condition for the system to take action is not comprehensive. A halting vehicle on the single neighboring lane is a common reason for oncoming vehicles to change

lanes and should be considered by the prediction to gain a safer margin.
While the delay of the prediction is compensated by a sufficient time margin in test case $\mathcal{A}$ due to a closer starting position and lower velocity of Vehicle #3, it cannot be compensated in $\mathcal{B}$ and $\mathcal{C}$, but are exposed as functional insufficiencies. It is apparent that parameterization is crucial to revealing the effect of a difficult environmental condition. This indicates the importance of exhaustively exploring the parameter space of a difficult environmental condition, in order to identify all critical cases. An exemplary method based on model checking [65] can be utilized. Since computational power is limited, our proposed method narrows the parameter space, thereby improving exploration efficiency.

Table 8.4.: Description of analyzed difficult environmental conditions and scenarios (©2025 SAE)

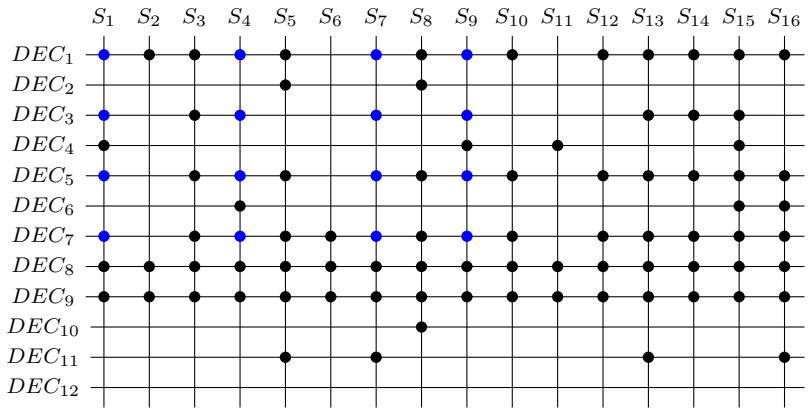
Difficult Environmental Conditions (DEC)			
DEC_1	Roadwork in the middle of the lane	DEC_7	The front cyclist drives on the road edge
DEC_2	Fully occluded relevant right-of-way sign	DEC_8	Heavy snow
DEC_3	An oncoming vehicle overtakes a halting vehicle	DEC_9	Faded road markings
DEC_4	The front vehicle in an unusual shape	DEC_{10}	Tandem cyclists with vehicle-like length
DEC_5	The front vehicle halts with a warning flasher on	DEC_{11}	Pedestrian near a tree
DEC_6	Missing lane marking on a turning lane	DEC_{12}	Snow covers the yellow lane marking

Reference Scenarios (S)			
S_1	Ego lane keeping with a vehicle broken down in front	S_9	Ego lane keeping with vehicle lane keeping in front
S_2	Ego lane keeping with the oncoming vehicle u-turning	S_{10}	Ego lane keeping with an oncoming vehicle lane keeping
S_3	Ego lane keeping at the junction with the traffic light	S_{11}	Ego lane changing left with a vehicle in front also changing
S_4	Ego turning right at the junction with the traffic light	S_{12}	Ego lane changing left with a vehicle in the target lane
S_5	Ego entering a roundabout with a pedestrian crossing in front	S_{13}	Ego lane keeping with pedestrian crossing from the right
S_6	Ego lane keeping with vehicle following closely behind	S_{14}	Ego lane keeping with an oncoming vehicle lane keeping
S_7	Ego approaching a crosswalk with a pedestrian on the sidewalk	S_{15}	Ego turning right at the junction with a vehicle in the target lane
S_8	Ego lane keeping with a cyclist crossing from off-road	S_{16}	Ego turning right with a pedestrian crossing straight

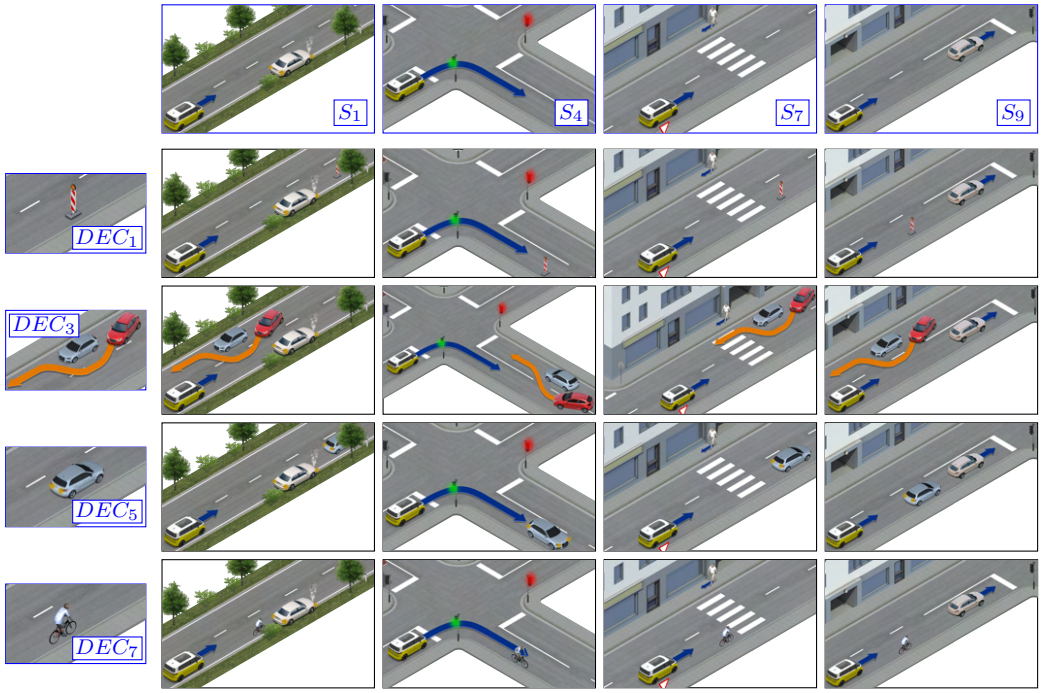
8.3.2. Extended Experiment with Multiple Scenarios

To show the applicability of our proposed method on a larger scale, we conduct the rule checking according to Table 8.1 based on 12 difficult environmental conditions and 16 scenarios from an industrial context. Table 8.4 exhibits the analyzed items, which cover diverse road geometry (e.g., straight roads and junctions), ego vehicle maneuvers (e.g., following lane, turning, and changing lane), and road user types (e.g., vehicles, pedestrians and cyclists). The rule checking stays on the semantic level, namely, we only qualitatively analyze which difficult environmental conditions can be combined with which scenario and derive corresponding samples.

We examine all 192 possible combinations (12×16). In this process, all analyzed difficult environmental conditions and scenarios can be mapped to one or multiple rules, according to the ODD elements or actions they require, allowing a conclusion about compatibility to



(a) Possible combinations



(b) 16 compatible combinations of scenarios injected with DECs

Figure 8.5.: Combination of difficult environmental conditions (DEC) and scenarios (S): exemplar combinations highlighted in blue (©2025 SAE)

be derived. Finally, 94 compatible pairs out of the 192 candidates are determined to generate potential critical scenarios. Figure 8.5 summarizes the results of the extended case study: Figure 8.5a provides an overview of the compatible pairs based on the analyzed difficult environmental conditions and the scenarios. Each intersection of a horizontal line (standing for a difficult environmental condition) and a vertical line (standing for a scenario) represents a rule

Table 8.5.: Description of exemplar combinations from Figure 8.5b (©2025 SAE)

Descriptions of Scenarios with Injected Difficult Environmental Conditions						
$S_1\times$	DEC_1	DEC_3	Ego lane keeping with a target vehicle broken down in front with ...	×	roadworks in front of the target vehicle	an oncoming vehicle overtakes a halting vehicle
	DEC_5	DEC_7			a halting vehicle with a warning flasher	cyclist driving on the road edge between them
$S_4\times$	DEC_1	DEC_3	Ego turning right at a traffic light controlled junction, encountering ...	×	roadworks	an oncoming vehicle overtakes a halting vehicle
	DEC_5	DEC_7			a halting vehicle with a warning flasher	cyclist driving on the road edge
$S_7\times$	DEC_1	DEC_3	Ego approaching crosswalk with target pedestrian on sidewalk with ...	×	roadworks blocking the ego lane	an oncoming vehicle overtakes a halting vehicle
	DEC_5	DEC_7			a halting vehicle with a warning flasher	cyclist driving on the road edge
$S_9\times$	DEC_1	DEC_3	Ego lane keeping with target vehicle lane keeping in front with ...	×	a beacon in lane	an oncoming vehicle overtakes a halting vehicle
	DEC_5	DEC_7			a halting vehicle with a warning flasher	cyclist driving on the road edge between them

checking process. A black dot is marked on the intersection when the pair is deemed compatible based on the process. Thus, 94 dots represent the identified compatible pairs. Among them, 16 pairs are randomly selected, highlighted in blue, and further visualized in Figure 8.5b to illustrate the generated critical scenarios. Accordingly, Table 8.5 lists a textual description of the 16 combinations.

8.4. Discussion and Limitations

Within the previous case study, we exhibit the applicability of the scenario generation method. To apply the method in an industrial context with a large amount of difficult environmental conditions and scenarios, the following limitations are discussed for further development:

Potential human bias in the designed rules. Detailed rules (cf. Table 8.1) are defined in our method to guide the searching and combining of compatible pairs of potential difficult environmental conditions and reference scenarios. In fact, these rules aim to maintain the essence of a scenario as much as possible when a difficult environmental condition is included. As the essence of a scenario is qualitatively defined based on expert knowledge, the corresponding rules can be biased. While we regard the introduction of such rules as a necessary initial step to structure the combination of difficult environmental conditions and scenarios, possible quantitative measurements should be considered as a complement. For instance, one can quantify the similarities between a reference scenario and the scenario including a difficult environmental condition and derive metrics to verify and improve the rules. Studies on scenario representativeness [22] investigate a similar direction.

Limited efficiency in manual semantic rule checking. The rule checking process in the provided case study (from the determination of suitable difficult environmental conditions to parameterization of the selected difficult environmental condition) is conducted manually. In an industrial context, huge amounts of potential difficult environmental conditions and reference scenarios are expected, and only a manual rule checking process will not be feasible

anymore. Thus, an automated rule checking is necessary. Most defined compatibility check rules require simply comparing ODD elements between the given scenario and the given potential difficult environmental condition. A formalized and machine-readable description of scenarios and difficult environmental conditions can facilitate the automation of such comparison. In this regard, the domain-specific languages by Bock et al. [13] and Zhu et al. [132] can be considered as a foundation, respectively. Meanwhile, variance control rules demand examining the influence of a potential difficult environmental condition on the ego vehicle and the existing dynamic objects in a reference scenario. To partially automate the variance control, the concept from Philipp et al. [82] can be implemented to derive relevant areas for the ego vehicle and other dynamic objects, respectively. In this way, the problem transfers to examining whether the difficult environmental condition related dynamic object can be placed in the relevant areas of the ego vehicle, while out of the areas of other existing dynamic objects.

9. Conformance, Compliance, and Traceability in the Test Implementation[1]

Industrial ADS development and verification is an iterative process, and maintaining traceability of difficult environmental conditions throughout this process is crucial for ensuring conformance with state-of-the-art standards and compliance with regulations. In this chapter, we firstly interpret relevant requirements from the international standard ISO 21448 [50] and the Commission Implementing Regulation EU 2022/1426 [27] and allocate difficult environmental conditions and corresponding activities within ADS's iterative development and verification process. Then, we discuss specific requirements for establishing traceability of environmental conditions during test activities. Finally, implementing these traceability requirements is demonstrated through a case study.

9.1. Requirements from ISO 21448

ISO 21448 mandates specific activities to be carried out throughout the iterative system development and V&V process (cf. [50, Clause 4, Figure 10]) to ensure the safety of the intended functionality. Among all required activities, those related to triggering conditions are particularly relevant for interpretation, due to the nature of difficult environmental conditions (cf. 3). Activities related to triggering conditions span across multiple phases of the iterative process and are involved in both theoretical analysis and practical testing

In the ADS design phase, potential triggering conditions[2] are analytically identified based on system specifications and the implementation [50, Clause 7]. In the early development iterations, the system design is relatively unformed, and a theoretical evaluation is able to expose some highly risky triggering conditions directly. For instance, an adopted object detection algorithm is known to have a high failure rate under rainy weather due to its mechanism. At the same time, the desired ODD comprises a region with large average annual rainfall. Experts can directly regard the rainy weather as an unacceptable triggering condition. Thus, a new development iteration due to function or ODD modifications can start without testing the system under rainy weather. With the rising maturity level of the system design, highly risky triggering conditions can be barely revealed with only theoretical analysis. A practical evaluation based on testing is necessary to investigate the concrete effects of potential triggering conditions as a part of the V&V strategy.

[1] This chapter is based on Paper IV [131] and therefore contains verbatim content previously published (©2025 SAE).

[2] The term *potential triggering condition* can be used when the ability to initiate a corresponding reaction is not yet established

In the phase of defining the V&V strategy [50, Clause 9], an overall validation target for the identified potential triggering conditions should be defined, e.g., predefined false positive and false negative rates are achieved after conducting all predefined test cases. The target should ultimately support the hypothesis that remaining unknown triggering conditions do not impose unreasonable risk. Besides, the validation target is suggested to be defined considering an appropriate development effort. [50, Clause 9] In the industrial development, it is unlikely to test all potential triggering conditions in an unlimited number of test cases due to limited time and resources. Instead, a feasible number n will be determined specifying the number of test cases to be implemented. A possible validation target could be that *all identified potential triggering conditions are tested within at least one test case, and the test result exhibits a sufficiently low cumulative rate of caused hazardous events.* In this case, the number n would be identical to the number of identified potential triggering conditions. Still, an overview of all possible test cases (featured with number m) based on identified potential triggering conditions facilitates establishing a feasible testing strategy: One can calculate a coverage index of testing as a ratio of actually implemented test cases to the complete set in each development iteration, namely n/m. The coverage index can contribute to either optimizing the testing in the next iterations or provide safety evidence in the final iteration. Once the validation target is established, selected n test cases should be implemented.

Finally, in the phase of testing [50, Clause 10], the ADS are tested with implemented and parameterized test cases containing potential triggering conditions. The system's performance is evaluated against predefined pass/fail criteria, e.g., whether a hazardous behavior of the ego vehicle is observed. When a test fails, the corresponding potential triggering condition is confirmed and documented as an actual triggering condition. Afterwards, related risks can be estimated considering the severity of the failure and the occurrence of the test scenarios. At the end of each iteration, the cumulative risks are compared to a predefined acceptance criterion [50, Clause 6]. Excessive risks require functional modifications based on the functional insufficiencies exposed by the triggering conditions and thus initiate a new development iteration.

9.2. Overall Evaluation Process

As introduced in the previous Chapter 8, generating potential critical scenarios based on difficult environmental conditions requires using reference scenarios. The EU type approval 2022/1426 [27] suggests employing various analytical frameworks, including ODD analysis and Object and Event Detection and Response (OEDR), to define nominal scenarios and ensure comprehensive coverage for specific applications. The type approval also specifies that the ADS shall be capable of performing the entire dynamic driving task under predefined nominal scenarios [27]. This indicates that nominal scenarios should be defined in the system specification phase and be tested in the early test phase. Therefore, it is logical to use nominal scenarios as reference scenarios for testing difficult environmental conditions: the normal operation of ADS should be assured in basic, nominal scenarios at first. Then, difficult environmental conditions are introduced to the already mastered nominal scenarios to generate potential critical scenarios.

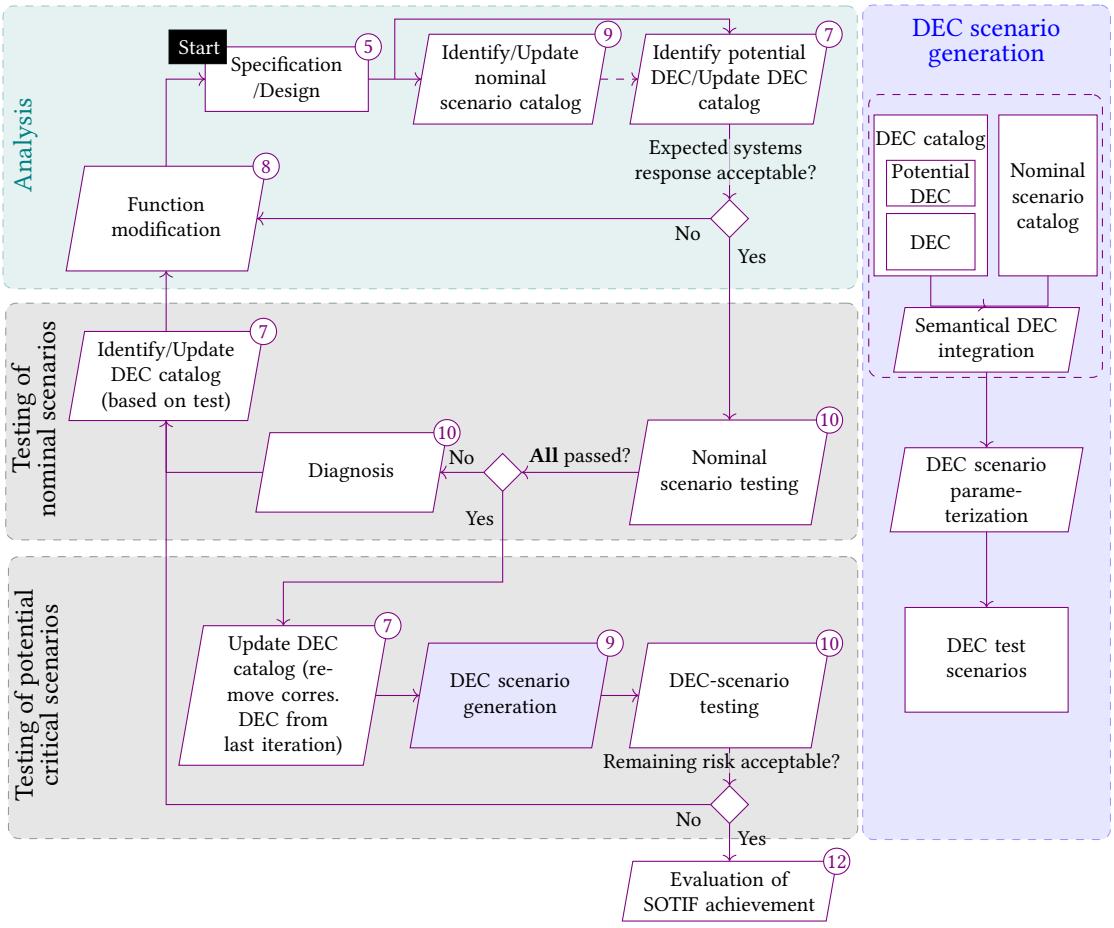


Figure 9.1.: Including difficult environmental conditions (DECs) into iterative development & test process of ADS: numbering of blocks refers to corresponding clauses in ISO 21448 (©2025 SAE)

In line with the activities outlined in ISO 21448 (cf. 9.1), the identification and testing of nominal scenarios and difficult environmental conditions can be positioned within the iterative ADS development and verification process. Figure 9.1 illustrates the overall process: during the system specification and design phase of the current ADS development iteration, nominal scenarios and potential difficult environmental conditions are thoroughly identified and cataloged. Once the requirement analysis phase concludes and the testing phase begins, nominal scenarios are tested first. It is important to note that the ADS could fail even in nominal scenarios, as these scenarios are just theoretically identified as easy situations for the system. In case the ADS fail a nominal scenario test, the nominal scenario could already contain an unknown difficult environmental condition. Any difficult environmental condition identified in an additional diagnosis procedure is added to the catalog of analytically defined conditions. At the same time, failure in a nominal scenario is considered unacceptable, prompting system upgrades through corresponding functional or ODD modifications. This triggers a new development iteration, which leads to updating the catalogs of nominal scenarios and difficult

environmental conditions. Once the ADS pass all predefined nominal scenario tests, the catalogs can be considered complete for that iteration. The next steps are to combine the identified difficult environmental conditions with the successfully tested nominal scenarios to derive test cases and conduct testing. ADS failures in tests involving different environmental conditions will indicate certain levels of risk. Any unacceptable risks, as determined by predefined acceptance criteria, will result in updating the rankings of difficult environmental conditions in the catalog and initiating ADS functional modifications. This starts a new iteration of development and verification. Conversely, if the remaining risks are deemed acceptable, the process can be concluded with the achievement of SOTIF.

9.3. Concrete Requirements for Establishing Traceability

To ensure traceability of difficult environmental conditions throughout the ADS development and verification process, it is essential to clarify how test cases should be managed, which intermediate test results need to be documented, and how the test outcomes from the current iteration can impact subsequent iterations. The following section outlines specific requirements based on these three key aspects.

9.3.1. Manage Test Cases on Three Abstraction Levels

As outlined in Chapter 8, difficult environmental conditions, nominal scenarios are combined and parameterized into potential critical scenarios across three abstraction levels [74]. Correspondingly, we suggest managing subsequently generated test cases on the same levels.

Functional level. Catalogs for difficult environmental conditions and nominal scenarios are established at the functional level. Similarly, a catalog of potential critical scenarios should be created at this level, reflecting the possible combinations of difficult environmental conditions and nominal scenarios. These catalogs together define the test space and help estimate subsequent testing efforts and resource allocation. For example, if 5000 nominal scenarios and 100 difficult environmental conditions are identified based on the ADS specification, and each environmental condition can be combined with 50 nominal scenarios, there would be a total of 10000 ($5000 + 100 * 50$) functional test cases for nominal scenarios and potential critical scenarios. Given a test capacity of one million test runs per verification iteration (subject to specific time and resource constraints), an average of 100 concrete test instances, with various parameterization, can be implemented for each functional test case[3].

Logical level. Nominal scenarios and potential critical scenarios should be parameterized and assigned with parameter ranges to generate logical test cases. A link should be established between logical test cases and their corresponding functional scenarios. Sampling methods (e.g., n-wise sampling) for each logical test case should be determined and documented. Over the course of the development and verification iterations, sampling methods for logical test

[3]For the sake of simplicity, in the example we consider using test capacity fully on testing of nominal scenarios and difficult environmental conditions. In reality, there can also be other scenario categories sharing the test resource, e.g., failure scenarios [27]

cases may evolve to explore the parameter space in different directions or generate additional samples.

Concrete level. Specific parameter values are determined through sampling methods to create concrete test cases. A link should be established between concrete test cases and their corresponding logical cases. The results of executing concrete test cases are documented at this level to determine whether the test passes or fails, assess the severity of failures, support function debugging, and enable regression testing. Key test results to be considered include:

- System version and test date
- Test pass/fail result
- In case of a failed test, the exact hazardous behaviors of ADS
- In case of the necessity of functional modifications, links to the functional modifications

To enable the reuse of concrete test cases, test results can be documented separately with a link to the test cases. Figure 9.2 demonstrates a possible data structure based on the introduced requirements to achieve traceability along the three abstraction levels.

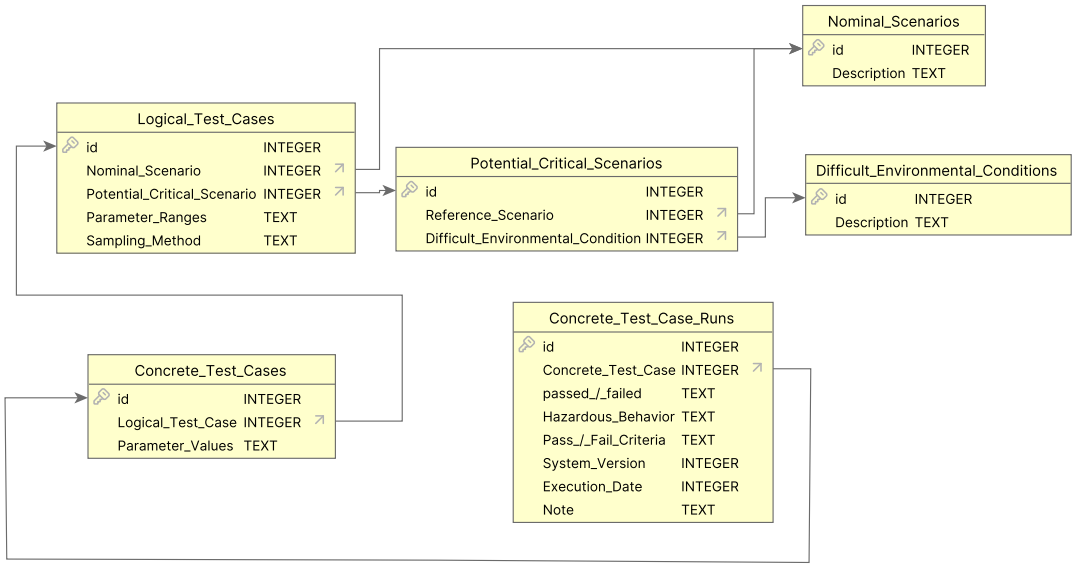


Figure 9.2.: Traceability in testing of nominal scenarios and difficult environmental conditions

9.3.2. Bringing Test Results into Decision-making

As it is previously assured that ADS can safely operate in the reference scenarios, ADS failing in a potential critical scenario can confirm the effectiveness of the corresponding difficult

environmental condition. Thus, the condition should be assigned a higher priority for focus in the subsequent test iteration. On the contrary, if the ADS pass all concrete test cases for a difficult environmental condition, it does not necessarily mean the condition is ineffective for the ADS, as the test cases are based on sampled parameters. The problematic parameter values may not have been exposed by the sampling method used. Therefore, the condition will remain in the catalog. In the next iteration, the condition may be assigned a lower priority or reparameterized with an updated sampling method.

When a difficult environmental condition is tested across multiple concrete test cases, it is possible that the ADS will pass some tests while failing others. In such cases, all test results should be analyzed in light of predefined acceptance criteria to determine whether a functional modification is necessary to address the condition. If a functional modification is made, the failed test cases should be retested for regression. If a functional modification is conducted, the failed concrete test cases should be tested again for regression. The possible test results for each difficult environmental condition and their implications for the next iteration are summarized in Table 9.1.

Table 9.1.: Possible test results regarding test cased of a difficult environmental condition and corresponding impact: TC - test case, DEC - difficult environmental condition

Test Result	Risk Assessment	Impact on System or Test Strategy			
		Func. modification	DEC Priority	Test Resource	Sampling method
All TCs passed	Acceptable	No	Decreased	Decreased	Unchanged
			Unchanged	Unchanged	Changed
At least one TC failed	Acceptable	No	Increased	Increased	Changed
			Unchanged	Unchanged	Changed
	Unacceptable	Yes	Increased	Increased	Changed

9.4. Technical Implementation of the Testing Process

To illustrate how the concept from the previous sections can be implemented, we reuse the difficult environmental conditions DEC_1, DEC_5, DEC_7, and reference scenarios S_4, S_7, S_9 from Table 8.4 to conduct a simplified case study. Let us assume the target ADS of automation level L4 [1] is specified for operating in the urban traffic in Germany. After analyzing the ADS specification and the ODD, a nominal scenario catalog is defined, consisting of

- S_4 (turning right at the junction),
- S_7 (approaching and crossing a crosswalk) and
- S_9 (keeping lane and maintaining distance to a lead vehicle)

Meanwhile, a difficult environmental condition catalog is identified as follows:

- DEC_1 Road works (Traffic beacon)

- DEC_5 Halting vehicle ahead with warning flasher
- DEC_7 Front cyclist on road edge

Based on the theoretical analysis of the catalogs, we proceed to testing. The three nominal scenarios are tested in multiple concrete variations, and the ADS performance is evaluated against the following intended functionality and behaviors:

- S4: detection of the traffic light status and keeping lane while turning right
- S7: detection of the crosswalk and pedestrians nearby and approaching the crosswalk with moderate speed (cf. German Road Traffic Act [33])
- S9: detection of a leading vehicle and adaptation of speed to maintain a safe distance (cf. EU regulation No.157 [116]), while keeping the lane

The ADS pass all test cases corresponding to these nominal scenarios, thus we move on to testing of potential critical scenarios based on difficult environmental conditions. According to the results of the compatibility check and variance control (cf. Table 8.1), all three difficult environmental conditions can be integrated in all three nominal scenarios, resulting in 9 potentially critical scenarios (cf. *Potential critical scenario catalog* in Figure 9.3). These 9 are tested in multiple concrete variations. The evaluation shows that the integrated difficult environmental conditions can be handled in most of the tested potential critical scenarios:

- Traffic beacons can be detected and are considered in path planning by making the ADS deviate a little to the left
- A halting vehicle ahead is detected, decelerated for, and cautiously passed
- A leading cyclist is detected, decelerated for and cautiously passed once possible with sufficient lateral distance (1.5 m)

However, specific instances of the combinations $S_4 \times DEC_1$ and $S_4 \times DEC_7$, namely the leading cyclist and the traffic beacon with the turn right scenario, lead to failures i.e., hazardous behaviors. In these instances, the positions of the traffic beacon or the cyclist are closer to the junction, and thus the ADS come close to them earlier compared to the parameterization in other instances. It indicates that the ADS could detect objects (that are smaller than cars) too late to react to them properly. In the scenario context of turning at a junction, the effect of a late detection is amplified and exposed as externally observable hazardous behaviors.

Both the cyclist and the traffic beacon are thus registered as confirmed difficult environmental conditions for the debugging phase and are marked with higher test priority for the next iteration. After the debugging, the ADS are modified by improving perception algorithms for object detection in the near range.

In the new iteration with the modified ADS function, no further nominal scenarios or difficult environmental conditions are identified, thus the corresponding catalogs stay the same. Repeating testing the new ADS version with nominal scenarios does not lead to failures. Subsequently, testing of potential critical scenarios with the previous combinations is performed

again. Due to the predefined higher priority, the combination of the right turn scenario with the cyclist and the traffic beacon is tested with more variations than the remaining combinations of the other two scenarios with the three difficult environmental conditions. This time, no hazardous behavior is exhibited in any concrete test cases. Thus, the iteration terminates with the achieved safety of the intended functionality.

Figure 9.3 provides a summary of the two iterations, highlighting key information. The figure is organized into three rows. The first row visualizes the nominal scenarios (S_4, S_7, S_9), difficult environmental conditions (DEC_1, DEC_5, and DEC_7), and potential critical scenarios based on their combinations. The second and third rows display individual test cases across two test iterations, corresponding to two consecutive system versions. The textual information to the right of each test case indicates the test result (pass or fail), the hazardous behavior, and the pass/fail criteria based on specific traffic rules or regulations. A summary of each iteration and the associated decisions is provided below the test cases.

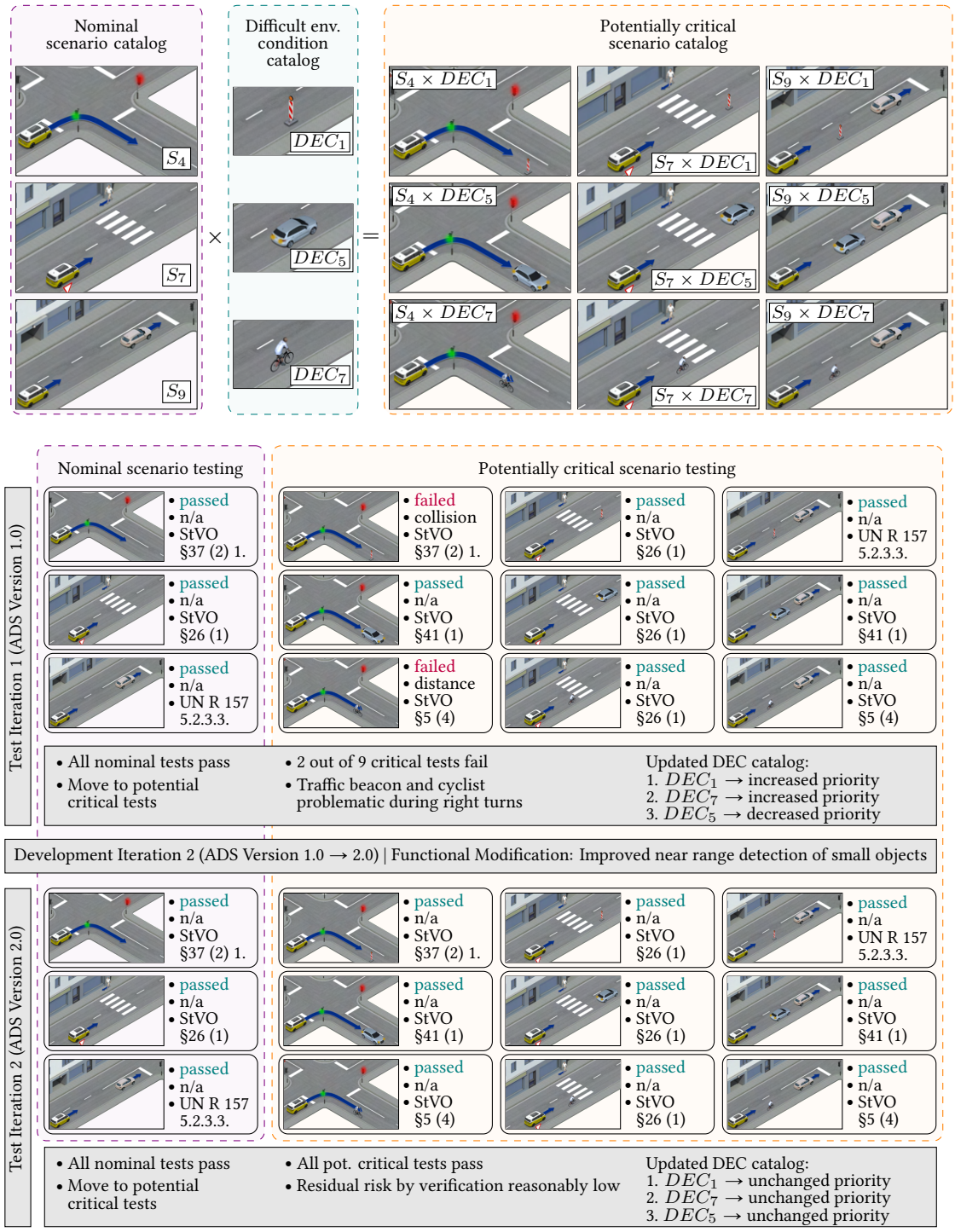


Figure 9.3.: Visualization of test and development activities in a hypothetical case study: n/a means no hazardous behavior exhibited in the testing

Part V.

Conclusion

10. Discussion and Reflection

In this chapter, we bring up the major findings in the thesis together and expound how they answer the five research questions proposed in Section 1.2. Following the contributions to each research question, we further discuss the open points and limitations in the findings. Therefore, the chapter is divided into five sections based on the research questions, each covering the contributions and limitations individually.

10.1. Understanding the Nature of Difficult Environmental Conditions

In order to approach the long tail problem in scenario-based testing, this thesis proposes a solution by exploring the operation environment of ADS and revealing the difficult environmental conditions. To set up a proper definition and scope for the exploration, it is vital to first understand what is actually a difficult environmental condition in terms of ADS. In this way, the first research question is described as:

***RQ 1**: How should the nature of difficult environmental conditions be understood?*

In Chapter 3, we first clarified the major difference and the relation between the environment and scenarios. We specified an environment to be everything lying outside a system's boundary. As for scenarios, we referred to the definition in the context of ADS development and verification. According to the most recognized definition by Ulbrich et al. [114], a scenario is the temporal development between multiple scenes in a sequence, while a scene represents a snapshot of the environment, encompassing both the scenery and dynamic elements, along with the self-representations of all actors and observers. We pointed out that the environment excludes the system itself, while scenarios comprise the system and its intended behaviors and functions towards its outer environment. At the same time, each scenario is designed to cover specific environmental conditions, and critical scenarios are designed to cover difficult environmental conditions. Based on this relation, we could firstly assume the reusability of the scenario modelling methods for modelling the environment and environmental conditions, which serve as a foundation to answer the following research questions RQ 3 and RQ 4. Secondly, we could further refer to the state-of-the-art literature regarding critical scenarios (including the definition or the methods for identifying critical scenarios) to learn what essentially makes a scenario critical for ADS, and thus analogously derive the requirements for distinguishing difficult environmental conditions from others.

In this way, we firstly analyzed from diverse academic works and the regulation EU 2022/1426 that critical scenarios are associated with ADS driving functions and their insufficiencies. Therefore, we further referred to the concept of *functional insufficiency* according to the ISO 21448

standard and arrived at two conclusions: (1) the essence of achieving the safety of the intended functionality (SOTIF) is continuously revealing and resolving unknown functional insufficiencies, until the remaining risk is acceptable, and (2) revealing unknown functional insufficiencies relies on discovering unknown hazardous (i.e., critical) scenarios via so-called *triggering conditions*. A triggering condition, according to the definition in ISO 21448, is "a specific condition of a scenario that serves as an initiator for a subsequent system reaction contributing to either a hazardous behavior or an inability to prevent or detect and mitigate a reasonably foreseeable indirect misuse". We observed the similarities in triggering conditions and difficult environmental conditions. By adopting the definition of triggering conditions partially and only focusing on the external factors of ADS, we could express the nature of difficult environmental conditions with the following requirements: firstly, difficult environmental conditions are deemed challenging for specific ADS functions and can potentially reveal functional insufficiencies. Secondly, difficult environmental conditions can result in hazardous behavior of ADS. Finally, difficult environmental conditions should serve the purpose of continuously exposing and resolving the weakness in ADS for achieving SOTIF. Hereby, the research question is answered.

10.2. Formalization of Difficult Environmental Conditions

As ADS are operating in an open world, their operational environment can contain infinite environmental conditions and thus a large amount of difficult environmental conditions. Meanwhile, collecting and utilizing difficult environmental conditions can involve many stakeholders. It is necessary to establish a formalization to avoid vagueness and duplication in the documentation. Aiming at a solution to this requirement, the second research question in the thesis is defined as:

***RQ 2**: How should difficult environmental conditions be formalized and managed?*

In Chapter 4, a sample set of difficult environmental conditions was collected from expert interviews and literature to develop a possible formalization. These difficult environmental conditions are summarized into three types: the *presence/absence*, the *behavior*, and the *interaction* of environmental factors. In the development of the formalization for each type, requirements for assuring the readability and maintenance are discussed. Each difficult environmental condition should be described concisely with the most essential information, while the description should be intuitive enough for the user to understand its meaning. In this way, the description units of a formalization are preliminarily determined as *Environmental Factor (EF)*, *Behavior (BHV)*, *Interaction (INTR)* for the essential, and *Attribute (ATTR)*, *Temporospatial Context (TSC)*, *Link to Function (L2F)* for the Intuitiveness. Moreover, natural language syntax is leveraged to develop relations and connections among the description units. As a result, description templates are provided for each type of difficult environmental conditions (cf. Figure 4.2). To enhance the formulation and avoid describing the same difficult environmental conditions in many variations, we applied certain linguistic constraints, such as forbidding the use of passive voice, disabling other tenses than the present tense, or specifying vocabulary for each description unit. The formalization is designed to be machine-readable, thus can be

extended or adapted to match with other domain-specific language (e.g., the language for the formalization of scenarios[13]). Specific description units can be utilized as sorting or filtering criteria to quickly group or search relevant difficult environmental conditions (e.g., to find all pedestrian-relevant difficult environmental conditions by sorting EF with the value *pedestrian*). Later in Part III, we collected a catalog of difficult environmental conditions with our proposed designated method on a larger scale. This directly served as a use case for the proposed formalization. It turned out that all the identified difficult environmental conditions can be covered by the predefined three types and formalized with the description units (cf. Appendix A). Therefore, the research question is fundamentally answered based on the aforementioned aspects.

Limitation. Still, there exist other possibilities to write the same identification in slightly different expressions. In some cases, a difficult environmental condition can be categorized as not only one type. For example, *Lying beacon on road* and *Beacon lies on road* are two variations of the same condition, mapping to two types. Such room for different interpretation is difficult to completely eliminated. So far, we have performed manual reviews and revisions to detect and merge the repetitions in the formalized results. We believe that this review and revision process can be supported and automated by applying techniques like Natural Language Processing (NLP), which is commonly used in the detection of similarities in text.

10.3. Identification of Difficult Environmental Conditions

With the understanding of the nature of difficult environmental conditions and how they can affect ADS to perform a hazardous behavior, the subsequent step is to design methods for identifying these conditions. This leads to the third research question of the thesis:

***RQ 3**: How can difficult environmental conditions be systematically identified?*

One contribution to answer this research question is provided in Chapter 5 with a knowledge-driven method, named Scenario-based Hazard and Fault Analysis (SHFA). SHFA guides experts to systematically analyze given scenarios and identify potential hazards and corresponding faults of ADS. In this way, the three aspects of the operational environment, the ADS, and the hazardous behavior are explicitly integrated in the method for the analysis, enabling a comprehensive identification of difficult environmental conditions. In the SHFA method, models and taxonomies are deployed to thoroughly decompose the scenario to environmental conditions, break down goals of *driving from A to B* to a sequence of intended maneuvers, deduce hazardous maneuvers, and organize ADS functions across multiple abstraction levels. The decomposition processes serve as a foundation to approach an exhaustive analysis and identification. Moreover, the formalized and constrained nature of the applied models and taxonomies helps eliminate human bias and supports the potential for automation.

Another contribution related to the research question is to demonstrate the applicability of the SHFA method via workshops with 16 domain experts (cf. Chapter 6). As a result, a catalog of 122 difficult environmental conditions covering sense, plan, and act components is established. The practical workshop experiments have confirmed the applicability of SHFA in an industrial context based on the large amount of identifications relative to a small application scale and the positive feedback from expert participants. Besides, we have learned lessons from

utilizing such analytical method on system level with experts with different backgrounds: On the one hand, function developers have the most experience and knowledge of concrete automated driving functions or components regarding their weaknesses and corresponding adverse situations. We observed that sensor experts are well-versed in identifying various types of failures in components such as LiDAR sensors, as well as common issues in sensor data processing (e.g., missing point clouds). However, they often focus on specific problems or exposure situations, which can lead them to deviate from a systematic analysis process. Additionally, they tend to operate at a more concrete functional level rather than considering the ADS as a whole. In contrast, test engineers and safety drivers are highly familiar with problematic scenarios based on their hands-on experience but may struggle to pinpoint root causes or link hazardous behaviors to specific system functions. On the other hand, requirement engineers typically have a strong grasp of overarching safety requirements, such as those derived from regulations or standards. They place particular emphasis on the completeness and thoroughness of the identification process. To fuse the strength of all these experts and elicit potential difficult environmental conditions with their experience and knowledge is still considered very beneficial, but at the same time not a trivial task. One promising strategy to enhance the efficiency of applying the SHFA method in expert workshops is to tailor the workshop structure to the participants' professional backgrounds, for example, by providing targeted guidance for specific steps or grouping participants with similar areas of expertise within the same session.

The last contribution to answer the research question is a data-driven approach to collect scenarios potentially containing difficult environmental conditions. In Chapter 7, we present a fully automatic pipeline to reconstruct ADS disengagement scenarios. The pipeline processes perception measurement data collected by automated driving prototype vehicles in urban areas. The pipeline is able to clean up the scenario by filtering out irrelevant objects, handling the dominant true positive inaccuracies, false positives, and false negatives in the data, and automatically generating a directly simulatable OpenSCENARIO file. By comparing the simulation with the video ground truth, we qualitatively showed that the generated virtual scenarios can present the essential parts of the disengagement records (cf. Table 7.4). A quantitative evaluation also indicated the fulfillment of individual intended functions of the pipeline. The reconstructed ADS disengagement scenarios can support the system developers in further identifying difficult environmental conditions in two possibilities. Firstly, by comparing the constructed and cleaned up scenarios with the original scenario based on the original perception measurements, objects causing major perception error can be obtained. These objects and their original attributes or behaviors in the original scenarios could be the difficult environmental conditions causing the disengagements, thus they are highly relevant for further investigation. Secondly, the disengagements could be caused by the overall traffic situation and were irrelevant to perception errors. Testing the constructed scenarios, which present the essential situations, can reveal difficult environmental conditions for the planning components of ADS.

Limitation. In terms of the knowledge-driven identification, the major limitation is on the human factors. Eliciting environmental causes for previously identified hazardous maneuvers highly depends on expert knowledge. Meanwhile, associating environmental conditions with specific system insufficiencies by experts is still performed in a qualitative manner in the scope

of this research. These factors still can lead to incompleteness and human bias in the result. Besides, no concrete ADS architecture and components are analyzed in the identification process. Therefore, the identification results provided in Appendix A may not be relevant for a certain system. For the data-driven approach, we stopped at potential critical scenarios, without identifying the related difficult environmental conditions due to no access to the original software stack used in the field experiments. Besides, there still exist some open points in the pipeline implementation. For example, completely missed objects, i.e., false negatives, cannot yet be recovered by the current pipeline. If such perception errors are encountered, the generated critical scenarios are not directly usable for the regression test. To cope with this problem, raw sensor data (e.g., camera images, point clouds) may need to be either reprocessed or manually labeled.

10.4. Converting Difficult Environmental Conditions into Scenarios

Difficult environmental conditions are intended for evaluating ADS and revealing their weaknesses. The evaluation is mainly implemented by testing ADS in the corresponding scenarios. Seeking a systematic method that converts difficult environmental conditions to scenarios, the fourth research question in this thesis is proposed as:

> ***RQ 4**: How should difficult environmental conditions be converted to potential critical scenarios?*

Chapter 2 discussed two possible approaches to derive potential critical scenarios based on difficult environmental conditions. One is to extend scenario context based on the given condition, the other is to combine the to be tested condition with already tested, hazard-free scenarios. We opted for the combination-based approach for two major arguments. Firstly, it enables the establishment of traceability between the test result and the difficult environmental condition. Secondly, it is more pragmatic within industrial development, as it allows developing scenarios and identifying difficult environmental conditions in parallel. For conducting the combination-based approach, it is crucial to figure out which scenarios are suitable to contain a given difficult environmental condition, while their essence is not significantly changed after the combination.

On the one hand, we defined two concepts, *compatibility check* and *variance control*, in Chapter 8 to guide the selection of suitable reference scenarios and restrict the modification of the scenarios. The compatibility check assures the availability of ODD elements or preconditions that are required by the difficult environmental conditions. The variance control avoids changing the reference scenario significantly during the combination process. For both, qualitative rules are defined based on expert knowledge. On the other hand, in the process of answering RQ 1, we determined that difficult environmental conditions are specific conditions in a scenario, and thus the methods for modeling scenarios can also be applied for modeling difficult environmental conditions and developing their test cases. Therefore, procedures in the combination-based approach are mapped to three abstraction levels according to the definition by Menzel et al. [74] as follows: On the functional level, compatibility check rules and a part

of variance control rules are implemented to initially determine suitable pairs of reference scenarios and difficult conditions. On the logical level, parameter ranges of reference scenarios are inherited, and the rest of the variance control rules are interpreted for deriving parameter ranges of difficult environmental conditions. On the concrete level, parameter values are sampled for difficult environmental conditions to finalize test case instances.

First, we conducted an end-to-end case study to illustrate how a pair of a difficult environmental condition and a reference scenario is determined and parameterized with the defined rules and procedures on the three abstraction levels. The reference scenario and the three generated test cases are tested in simulation as a proof-of-concept. We further analyzed the test result to illustrate the impact of different parameterization of a difficult environmental condition and to demonstrate revealing corresponding functional insufficiencies. Besides, we performed an extended case study with 12 difficult environmental conditions and 16 reference scenarios on the functional level and confirmed the applicability of our proposed rules on a larger scale. Thereby, the research question is answered with the proposed combination-based approach, the defined rules, the structured scenario generation on diverse abstraction levels, and the proof-of-concept case studies.

Limitation. As discussed in Section 8.4, we rely on expert interpretation to define what is essential and not modifiable, and what can be modified to what extent in a scenario for conducting the compatibility check and variance control. These interpretations are qualitative, so the corresponding rules could be subject to human bias and are incomplete. To cope with these problems, the rules could be further extended or broken down, and should be enhanced with quantitative rationales. Besides, automation techniques should be considered in the conduction of the compatibility check and the variance control to improve the efficiency in the search for reference scenarios for difficult environmental conditions.

10.5. Establishment of Traceability, Conformance, and Compliance

Considering industrial development and verification processes of ADS, the system specification and ODD definition can be adapted frequently due to both technical reasons (e.g., a discovered functional insufficiency requires ADS functional modification) and non-technical reasons (e.g., changing of target market due to the occupancy by other competitors results in updates of ODD). Accordingly, the identification and testing activities of difficult environmental conditions need to be aligned with the ADS development and verification process. In this regard, to establish the traceability of the intermediate identifications and testing results and exhibit compliance with industrial standards and regulations is of great significance. Thus, the last research question is formulated as:

> ***RQ 5**: How can the traceability of difficult environmental conditions be established in the development lifecycle of ADS with conformance to state-of-the-art standards and compliance with regulations?*

As elaborated in Chapter 3, difficult environmental conditions largely adopt the definition of triggering conditions from ISO 21448. Therefore, we collected the descriptions regarding

triggering conditions through all clauses of ISO 21448 and interpreted them into required activities. These activities range from identification to the evaluation and finally to the analysis and documentation of evaluation results. Meanwhile, we referred to the concept of nominal scenarios in the EU regulation EU 2022/1426. We explained why nominal scenarios are suitable to be utilized as reference scenarios for conducting the combination-based scenario generation (cf. Chapter 8). Subsequently, we applied the requirements towards triggering conditions to difficult environmental conditions. We allocated these with the identification and testing of nominal scenarios in a SOTIF-oriented, iterative ADS development and verification process. In this way, the activities related to difficult environmental conditions introduced in this thesis conform to both ISO 21448 and EU 2022/1426, answering one part of the research question.

Afterwards, we proposed concrete requirements to establish traceability of difficult environmental conditions in the illustrated process. We clarified how test cases should be managed on functional, logical, and concrete levels, which intermediate test results should be documented, and how the test results from the current iteration can influence the next iteration. We demonstrated a simplified case study based on a hypothetical ADS, following the proposed SOTIF-oriented test process (cf. Figure 9.1). Difficult environmental conditions are tested, evaluated, and updated in the case study. Based on the test result, the ranking of difficult environmental conditions was updated in the catalog: those that led to failed tests with ADS hazardous behaviors are assigned with higher priority, namely, more test runs for the next iteration, while those not yet reveal any system insufficiencies are either assigned with less priority or new sampling methods for further investigation with different test cases. Based on the elaboration of these concepts and the case study, the other part of the research question is deemed answered.

Limitation. The thesis presented only a proof-of-concept with a small-scale case study for the proposed SOTIF-oriented process. However, the process was not validated on a large scale. There can therefore be three major potential difficulties for applying the process in real industrial projects. First, the process requires testing nominal scenarios prior to testing any critical scenarios. In real-world implementation, this requirement may not always be fulfilled due to resource allocation or management-level strategies. Secondly, the process requires to always restart a new iteration of system design if any nominal scenario test is failed. This requires tight cooperation between the verification and the development departments (e.g., everything is developed in house), which can be challenging for the commercial model, where two different companies or organizations are responsible for the ADS development and the system integration and homologation separately. Lastly, the strategy and requirement to combine difficult environmental conditions with all compatible nominal scenarios for generating test cases provides a solution for approaching completeness. However, this will also likely lead to an exponential number of test cases. A further strategy to determine a feasible yet sufficient subset of the test cases should be developed to cope with the problem.

11. Conclusion and Outlook

In this chapter, we summarize the major contributions in the thesis as proof of achieving the predefined research goal. We also provide recommendations for future works based on this research.

11.1. Conclusion

The long-tail problem in scenario-based testing poses a significant challenge to the safety verification of Automated Driving Systems (ADS). Addressing this issue requires the effective identification and testing of critical scenarios. To achieve the comprehensiveness, three key dimensions of ADS must be considered: system functions and components, potential hazardous behaviors, and operational environments. In response to this need, this thesis introduces the concept of *difficult environmental conditions*. Unlike many existing approaches focusing on generating critical scenarios for isolated driving functions or components, the proposed concepts and methods target system-level verification. It culminates in a coherent framework for identifying, formalizing, testing, and tracing difficult environmental conditions throughout the ADS development lifecycle.

Building on terms like *triggering condition* and *functional insufficiencies* from the ISO 21448 standard, we defined difficult environmental conditions as specific conditions in the operational environment of ADS that can trigger one or multiple functional insufficiencies, ultimately resulting in hazardous behaviors of the ADS. From this understanding, we categorized difficult environmental conditions into three types and created formalized, machine-readable descriptions for documentation and management. We developed the Scenario-based Hazard and Fault Analysis (SHFA) method to systematically identify these conditions. Through a series of workshops with domain experts, we established the first catalog of 122 difficult environmental conditions. These results were standardized using our proposed formalization templates. Additionally, we enhanced the identification process with a data-driven approach by developing an automatic pipeline to post-process perception measurement data and reconstruct disengagement scenarios from automated vehicle prototypes. These reconstructed scenarios either contained difficult environmental conditions for ADS planning components or can be further analyzed with ground truth data to identify conditions affecting perception components. Lastly, we demonstrated how critical scenarios can be derived from difficult environmental conditions for testing. We defined qualitative rules to find compatible nominal scenarios for a specific difficult environmental condition, so that they can be combined and parameterized to generate concrete test cases. On the other hand, we determined the positions for identifying and testing difficult environmental conditions and nominal scenarios in the iterative ADS development and verification process. To do so, we referred to the standard ISO 21448 and the EU

type approval EU 2022/1426, so that the designed process also fulfills their objectives and requirements. Based on the process, we defined concrete requirements to establish traceability of difficult environmental conditions in testing.

As with other research in the field of safety verification and validation of ADS, this work faced the enduring challenge of demonstrating completeness and sufficiency. To address this challenge in the context of difficult environmental conditions, this thesis adopted a multifaceted approach along three key dimensions. First, the identification process considered ADS functions holistically, encompassing the complete sense–plan–act chain, and applied the systematic SHFA method to ensure exhaustive scenario analysis. Second, both knowledge-driven and data-driven techniques were employed to mitigate the limitations and biases inherent in relying solely on one approach. Third, a systematic method was introduced to comprehensively generate all candidate test scenarios based on the previously identified difficult environmental conditions, thereby supporting thorough coverage in subsequent testing efforts. In parallel with the contributions of this thesis, we also recognized the importance of identifying a complete set of nominal scenarios in accordance with EU Regulation 2022/1426. Nominal scenarios not only serve as a critical input for identifying difficult environmental conditions with the SHFA method, but also play an essential role in the subsequent generation of test cases. Advancing this complementary line of research is expected to support further efforts toward achieving completeness in scenario-based ADS validation.

11.2. Recommendations for Future Work

Following the wish according to a traditional Chinese idiom “to toss a brick to attract the jade”[1], we tossed “the brick” by introducing the concept of difficult environmental conditions, proposing research questions and the corresponding solutions. In this section, we aim to attract ”the jade”, i.e., more research into difficult environmental conditions, by discussing open questions and future directions. In Chapter 10, we already discussed the insufficiencies related to the solutions to the research questions RQ 1 to RQ 5. These gaps provide opportunities for refining the proposed solutions. Besides, the following topics pertaining to difficult environmental conditions are not yet discussed or not the focus of this thesis, but should be considered:

Managing diverse concreteness in the identifications. Although difficult environmental conditions are primarily all identified and documented on the functional level, we observed varying levels of concreteness at the micro level (cf. Catalog in Appendix A). For example, the entry *Faded yellow lane markings* is more concrete than the entries *Unusual negation on traffic sign* and *Object with thin profile*. More concrete entries are generally more intuitive for tasks like deriving coverage indices or designing test cases. While we aimed to incorporate expert knowledge in identifying concrete conditions, overfitting may occur if every entry is overly specific. Therefore, criteria should be defined for determining the optimal level of concreteness during the identification and documentation process.

Theoretical evaluation and risk analysis. Our evaluation of difficult environmental conditions focused on test case generation and process design. However, ISO 21448 suggests that

[1] “抛砖引玉”, pāo zhuān yǐn yù [123]

theoretical evaluation should precede the testing phase. Additionally, residual risks from difficult environmental conditions that are tested but not fully addressed should be assessed using a specific methodology. A potential starting point for this analysis is applying exposure and severity analysis, similar to the ASIL analysis in ISO 26262.

Traceability and linkage to other work products. As difficult environmental conditions are a novel concept in this thesis, existing V&V strategies may already address some aspects (e.g., comprehensive scenario catalogs) to ensure completeness or manage long-tail problems. However, incorporating difficult environmental conditions into the V&V strategy can further structure, facilitate, and enhance the V&V process and help demonstrate conformance with standards and compliance to regulations for homologation. To this end, we recommend mapping difficult environmental conditions to other implemented requirements or measures, such as scenario databases, hazard and risk analysis, and ODD taxonomy.

Combinatorial testing with multiple conditions. We explained the overall testing method and process of difficult environmental conditions in Chapter 8 and Chapter 9. For the sake of simplicity and keeping the main focus in the thesis, we suggested considering one difficult environmental condition in a test case at a time and did not discuss the combination of multiple conditions in one test case. Nevertheless, we consider applying combinatorial testing as meaningful in some situations. For example, if several difficult environmental conditions have not revealed any functional insufficiencies in multiple system verification iterations, they can be combined to derive new test cases.

Bibliography

[1] 2021. *SAE J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. https://www.sae.org/standards/content/j3016_202104/

[2] Ahmad Adee, Roman Gansch, Peter Liggesmeyer, Claudius Glaeser, and Florian Drews. 2021. Discovery of Perception Performance Limiting Triggering Conditions in Automated Driving. In *2021 5th International Conference on System Reliability and Safety (ICSRS)*. IEEE, Palermo, Italy, 248–257. https://doi.org/10.1109/ICSRS53853.2021.9660641

[3] Matthias Althoff, Markus Koschi, and Stefanie Manzinger. 2017. CommonRoad: Composable benchmarks for motion planning on roads. In *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Los Angeles, CA, USA, 719–726. https://doi.org/10.1109/IVS.2017.7995802

[4] Matthias Althoff and Sebastian Lutz. 2018. Automatic Generation of Safety-Critical Test Scenarios for Collision Avoidance of Road Vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Changshu, China, 1326–1333. https://doi.org/10.1109/IVS.2018.8500374

[5] Christian Thomas Amersbach. 2020. *Functional Decomposition Approach-Reducing the Safety Validation Effort for Highly Automated Driving*. Ph. D. Dissertation. Technische Universität Darmstadt.

[6] Mikael Andersson, Irene Natale, Andreas Tingberg, and Jakob Kaths. 2022. scenariogeneration. https://github.com/pyoscx/scenariogeneration

[7] ASAM. 2024. ASAM OpenDRIVE®. https://www.asam.net/standards/detail/opendrive/ (last accessed on 25.01.2025).

[8] ASAM. 2024. ASAM OpenSCENARIO® XML. https://www.asam.net/standards/detail/openscenario-xml/ (last accessed on 25.01.2025).

[9] Baidu Apollo Team. 2017. Apollo: Open Source Autonomous Driving. https://github.com/ApolloAuto/apollo last accessed on 15.04.2024.

[10] Daniel Baumann, Raphael Pfeffer, and Eric Sax. 2021. Automatic Generation of Critical Test Cases for the Development of Highly Automated Driving Functions. In *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*. IEEE, Helsinki, Finland. https://doi.org/10.1109/VTC2021-Spring51267.2021.9448686

[11] Maximilian Bäumler and Günther Prokop. 2022. Generating ADS Test Scenarios from Police Accident Data: How to Predict the Type of Road Traffic Accident Accurately. (2022). https://doi.org/10.2139/ssrn.4295798

[12] Rodrigo F. Berriel, Edilson de Aguiar, Alberto F. de Souza, and Thiago Oliveira-Santos. 2017. Ego-Lane Analysis System (ELAS): Dataset and algorithms. *Image and Vision Computing* 68 (2017), 64–75. https://doi.org/10.1016/j.imavis.2017.07.005

[13] Florian Bock, Christoph Sippl, Aaron Heinz, Christoph Lauer, and Reinhard German. 2019. Advantageous Usage of Textual Domain-Specific Languages for Scenario-Driven Development of Automated Driving Functions. In *2019 IEEE International Systems Conference (SysCon)*. IEEE, Orlando, FL, USA, 1–8. https://doi.org/10.1109/SYSCON.2019.8836912

[14] Alexandra M. Boggs, Ramin Arvin, and Asad J. Khattak. 2020. Exploring the who, what, when, where, and why of automated vehicle disengagements. *Accident Analysis & Prevention* 136 (2020), 105406. https://doi.org/10.1016/j.aap.2019.105406

[15] Jasmin Breitenstein, Jan-Aike Termohlen, Daniel Lipinski, and Tim Fingscheidt. 2020. Systematization of Corner Cases for Visual Perception in Automated Driving. In *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Las Vegas, NV, USA. https://doi.org/10.1109/IV47402.2020.9304789

[16] Andreas Bussler, Lukas Hartjen, Robin Philipp, and Fabian Schuldt. 2020. Application of Evolutionary Algorithms and Criticality Metrics for the Verification and Validation of Automated Driving Systems at Urban Intersections. In *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Las Vegas, NV, USA, 128–135. https://doi.org/10.1109/IV47402.2020.9304662

[17] Angelo Coppola, Claudio D'Aniello, Luigi Pariota, and Gennaro Nicola Bifulco. 2023. Assessing safety functionalities in the design and validation of driving automation. *Transportation Research Part C: Emerging Technologies* 154 (2023), 104243. https://doi.org/10.1016/j.trc.2023.104243

[18] Frank Crawley and Brian Tyler. 2015. *HAZOP: Guide to best practice*. Elsevier. https://doi.org/10.1016/C2014-0-04859-9

[19] Clement Creusot and Asim Munawar. 2015. Real-time small obstacle detection on highways using compressive RBM road reconstruction. In *2015 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Seoul, Korea (South). https://doi.org/10.1109/IVS.2015.7225680

[20] Breno Dantas Cruz, Bargav Jayaraman, Anurag Dwarakanath, and Collin McMillan. 2017. Detecting Vague Words & Phrases in Requirements Documents in a Multilingual Environment. In *2017 IEEE 25th International Requirements Engineering Conference (RE)*. IEEE, Lisbon, Portugal, 233–242. https://doi.org/10.1109/RE.2017.24

[21] Gabriel Rodrigues De Campos, Roozbeh Kianfar, and Mattias Brännström. 2021. Precautionary Safety for Autonomous Driving Systems: Adapting Driving Policies to Satisfy Quantitative Risk Norms. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, Indianapolis, IN, USA, 645–652. https://doi.org/10.1109/ITSC48978.2021.9564879

[22] Erwin de Gelder, Jasper Hof, Eric Cator, Jan-Pieter Paardekooper, Olaf Op den Camp, Jeroen Ploeg, and Bart de Schutter. 2022. Scenario Parameter Generation Method and Scenario Representativeness Metric for Scenario-Based Assessment of Automated Vehicles. *IEEE Transactions on Intelligent Transportation Systems* 23, 10 (2022), 18794–18807. https://doi.org/10.1109/TITS.2022.3154774

[23] Department for Transport. 2024. A vision for GB type approval. https://www.gov.uk/government/publications/a-vision-for-gb-type-approval/a-vision-for-gb-type-approval (last accessed on 01.05.2025).

[24] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An Open Urban Driving Simulator. arXiv:1711.03938 [cs.LG] https://arxiv.org/abs/1711.03938

[25] Clifton A. Ericson. 2005. *Hazard Analysis Techniques for System Safety*. John Wiley & Sons. 10–12 pages. https://doi.org/10.1002/0471739421

[26] E. Esenturk, S. Khastgir, A. Wallace, and P. Jennings. 2021. Analyzing Real-world Accidents for Test Scenario Generation for Automated Vehicles. In *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Nagoya, Japan, 288–295. https://doi.org/10.1109/IV48863.2021.9576007

[27] European Commission. 2022. Commission Implementing Regulation (EU) 2022/1426 of 5 August 2022 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 (Text with EEA relevance). http://data.europa.eu/eli/reg_impl/2022/1426/oj (last accessed on 26.09.2023).

[28] Víctor J. Expósito Jiménez, Helmut Martin, Christian Schwarzl, Georg Macher, and Eugen Brenner. 2022. Triggering Conditions Analysis and Use Case for Validation of ADAS/ADS Functions. In *Computer Safety, Reliability, and Security. SAFECOMP 2022 Workshops*. Springer International Publishing, 11–22. https://doi.org/10.1007/978-3-031-14862-0_1

[29] Paolo Falcone, Francesco Borrelli, Jahan Asgari, H. Eric Tseng, and Davor Hrovat. 2007. A model predictive control approach for combined braking and steering in autonomous vehicles. In *2007 Mediterranean Conference on Control & Automation*. IEEE, Athens, Greece, 1–6. https://doi.org/10.1109/MED.2007.4433694

[30] Marta V Faria, Patrícia C Baptista, Tiago L Farias, and João MS Pereira. 2018. Assessing the impacts of driving environment on driving behavior patterns. *Transportation* 47, 3 (2018), 1311–1337. https://doi.org/10.1007/s11116-018-9965-5

[31] Francesca Favarò, Sky Eurich, and Nazanin Nader. 2018. Autonomous vehicles' disengagements: Trends, triggers, and regulatory limitations. *Accident Analysis & Prevention* 110 (2018), 136–148. https://doi.org/10.1016/j.aap.2017.11.001

[32] Federal Ministry for Digital and Transport (Germany). 2022. Verordnung zur Regelung des Betriebs von Kraftfahrzeugen mit automatisierter und autonomer Fahrfunktion und zur Änderung straßenverkehrsrechtlicher Vorschriften. https://www.bundesrat.de/bv.html?id=0086-22 (last accessed on 01.05.2025).

[33] Federal Ministry of Justice and Consumer Protection (Germany). 2013. Road Traffic Ordinance (StVO). Online. (2013).

[34] Daniel J Fremont, Edward Kim, Tommaso Dreossi, Shromona Ghosh, Xiangyu Yue, Alberto L Sangiovanni-Vincentelli, and Sanjit A Seshia. 2022. Scenic: a language for scenario specification and data generation. *Machine Learning* 112, 10 (2022), 3805–3849. https://doi.org/10.1007/s10994-021-06120-5

[35] Alessio Gambi, Tri Huynh, and Gordon Fraser. 2019. Generating effective test cases for self-driving cars from police reports. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE '19)*. ACM, 257–267. https://doi.org/10.1145/3338906.3338942

[36] German Aerospace Center. [n. d.]. PEGASUS Research Project. https://www.pegasusprojekt.de last accessed on 15.04.2024.

[37] German Federal Motor Transport Authority. 2025. EU-Typgenehmigungen für Kraftfahrzeuge mit autonomer Fahrfunktion in Kleinserie. https://www.kba.de/DE/Themen/Typgenehmigung/Autonomes_automatisiertes_Fahren/EU_Typgenehmigung/eu_typgenehmigung_node.html (last accessed on 01.05.2025).

[38] German Federal Motor Transport Authority. 2025. Nationale Betriebserlaubnis für Kraftfahrzeuge mit autonomer Fahrfunktion. https://www.kba.de/DE/Themen/Typgenehmigung/Autonomes_automatisiertes_Fahren/nationale_Betriebserlaubnis/nationale_betriebserlaubnis_node.html (last accessed on 01.05.2025).

[39] Philipp Glauner, Axel Blumenstock, and Martin Haueis. 2012. Effiziente Felderprobung von Fahrerassistenzsystemen. In *8. Uni-DAS e.V. Workshop Fahrerassistenz*. 5–14.

[40] Robert Graubohm, Marvin Loba, Marcus Nolte, and Markus Maurer. 2023. Identifikation auslösender Umstände von SOTIF-Gefährdungen durch systemtheoretische Prozessanalyse. *at - Automatisierungstechnik* 71, 3 (2023), 209–218. https://doi.org/10.1515/auto-2022-0164

[41] Rohan Gudla, Vijay Shankar Telidevulapalli, Jayasree Sarada Kota, Gayathri Mandha, et al. 2022. Review on self-driving cars using neural network architectures. *World Journal of Advanced Research and Reviews* 16, 2 (2022), 736–746. https://doi.org/10.30574/wjarr.2022.16.2.1240

[42] Magnus Gyllenhammar, Gabriel Rodrigues de Campos, and Martin Törngren. 2025. The Road to Safe Automated Driving Systems: A Review of Methods Providing Safety Evidence. *IEEE Transactions on Intelligent Transportation Systems* 26, 4 (2025), 4315–4345. https://doi.org/10.1109/TITS.2025.3532684

[43] Mohammed Hadi, Prasoon Sinha, and John R. Easterling. 2007. Effect of Environmental Conditions on Performance of Image Recognition-Based Lane Departure Warning System. *Transportation Research Record: Journal of the Transportation Research Board* 2000, 1 (2007), 114–120. https://doi.org/10.3141/2000-14

[44] Lukas Hartjen. 2023. *Semantic Classification of Urban Traffic Scenarios for the Validation of Automated Driving Systems.* Ph. D. Dissertation. Technische Universität Braunschweig.

[45] Lukas Hartjen, Robin Philipp, Fabian Schuldt, Falk Howar, and Bernhard Friedrich. 2019. Classification of Driving Maneuvers in Urban Traffic for Parametrization of Test Scenarios. In *9. Tagung Automatisiertes Fahren.* Lehrstuhl für Fahrzeugtechnik mit TÜV SÜD Akademie, Munich, Germany. https://mediatum.ub.tum.de/doc/1535131/file.pdf

[46] Florian Hauer, Tabea Schmidt, Bernd Holzmüller, and Alexander Pretschner. 2019. Did We Test All Scenarios for Automated and Autonomous Driving Systems?. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC).* IEEE, Auckland, New Zealand, 2950–2955. https://doi.org/10.1109/ITSC.2019.8917326

[47] An Huang, Xingyu Xing, Tangrui Zhou, and Junyi Chen. 2021. A Safety Analysis and Verification Framework for Autonomous Vehicles Based on the Identification of Triggering Events. In *SAE Technical Paper Series (ASSTC).* SAE International. https://doi.org/10.4271/2021-01-5010

[48] IEC 60812:2018. 2018. *Failure modes and effects analysis (FMEA and FMECA).* Standard. Geneva, Switzerland.

[49] IEC 61025:2006. 2006. *Fault tree analysis (FTA).* Standard. Geneva, Switzerland.

[50] ISO 21448:2022. 2022. *Road vehicles — Safety of the intended functionality.* Standard. Geneva, Switzerland.

[51] ISO 26262:2018. 2018. *Road vehicles — Functional safety.* Standard. Geneva, Switzerland.

[52] ISO 34502:2022. 2022. *Road vehicles — Test scenarios for automated driving systems — Scenario based safety evaluation framework.* Standard. Geneva, Switzerland.

[53] ISO 34503:2023. 2023. *Road vehicles — Test scenarios for automated driving systems — Specification for operational design domain.* Standard. Geneva, Switzerland.

[54] ISO 34504:2024. 2024. *Road vehicles — Test scenarios for automated driving systems - Scenario categorization.* Standard. Geneva, Switzerland.

[55] ISO/IEC/IEEE 29119:2022. 2022. *Software and systems engineering — Software testing.* Standard. Geneva, Switzerland.

[56] ISO/PAS 21448:2019. 2019. *Road vehicles — Safety of the intended functionality.* Standard. Geneva, Switzerland.

[57] Kalena Thomhave (Automotive Dive). 2025. Volkswagen and Uber to test, deploy robotaxis. https://www.automotivedive.com/news/Volkswagen-uber-partnership-robotaxi-service/747071/ (last accessed on 12.05.2025).

[58] Nidhi Kalra and Susan M. Paddock. 2016. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice* 94 (2016), 182–193. https://doi.org/10.1016/j.tra.2016.09.010

[59] Takeo Kanade, Chuck Thorpe, and William Whittaker. 1986. Autonomous land vehicle project at CMU. In *Proceedings of the 1986 ACM fourteenth annual conference on Computer science - CSC '86 (CSC '86)*. ACM Press, 71–80. https://doi.org/10.1145/324634.325197

[60] Dhanoop Karunakaran, Julie Stephany Berrio, Stewart Worrall, and Eduardo Nebot. 2022. Automatic lane change scenario extraction and generation of scenarios in OpenX format from real-world data. arXiv:2203.07521 [cs.RO] https://arxiv.org/abs/2203.07521

[61] Siddartha Khastgir, Simon Brewerton, John Thomas, and Paul Jennings. 2021. Systems Approach to Creating Test Scenarios for Automated Driving Systems. *Reliability Engineering & System Safety* 215 (2021), 107610. https://doi.org/10.1016/j.ress.2021.107610

[62] Kirsten Korosec (TechCrunch). 2025. Waymo has doubled its weekly robotaxi rides in less than a year. https://techcrunch.com/2025/02/27/waymo-has-doubled-its-weekly-robotaxi-rides-in-less-than-a-year/ (last accessed on 12.05.2025).

[63] Moritz Klischat, Edmond Irani Liu, Fabian Holtke, and Matthias Althoff. 2020. Scenario Factory: Creating Safety-Critical Traffic Scenarios for Automated Vehicles. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, Rhodes, Greece, 1–7. https://doi.org/10.1109/ITSC45102.2020.9294629

[64] Emil Knabe et al. 2021. Environment simulator minimalistic (esmini). *Accessed on* 20 (2021).

[65] Lukas König, Christian Heinzemann, Alberto Griggio, Michaela Klauck, Alessandro Cimatti, Franziska Henze, Stefano Tonetta, Stefan Küperkoch, Dennis Fassbender, and Michael Hanselmann. 2024. Towards Safe Autonomous Driving: Model Checking a Behavior Planner during Development. In *Tools and Algorithms for the Construction and Analysis of Systems*. Springer Nature Switzerland, 44–65. https://doi.org/10.1007/978-3-031-57249-4_3

[66] Philip Koopman, Beth Osyk, and Jack Weast. 2019. Autonomous Vehicles Meet the Physical World: RSS, Variability, Uncertainty, and Proving Safety. In *Computer Safety, Reliability, and Security. SAFECOMP 2019*. Springer International Publishing, 245–253. https://doi.org/10.1007/978-3-030-26601-1_17

[67] Birte Kramer, Christian Neurohr, Matthias Büker, Eckard Böde, Martin Fränzle, and Werner Damm. 2020. Identification and Quantification of Hazardous Scenarios for Automated Driving. In *Model-Based Safety and Assessment*. Springer International Publishing, 163–178. https://doi.org/10.1007/978-3-030-58920-2_11

[68] Nancy G. Leveson and John P. Thomas. 2018. STPA handbook. (2018). https://psas.scripts.mit.edu/home/get_file.php?name=STPA_handbook.pdf (last accessed on 17.12.2024).

[69] Hailiang Li, Bin Zhang, Yu Zhang, Xilin Dang, Yuwei Han, Linfeng Wei, Yijun Mao, and Jian Weng. 2021. A defense method based on attention mechanism against traffic sign adversarial samples. *Information Fusion* 76 (2021), 55–65. https://doi.org/10.1016/j.inffus.2021.05.005

[70] Yihao Li, Jianbo Tao, and Franz Wotawa. 2020. Ontology-based test generation for automated and autonomous driving functions. *Information and Software Technology* 117 (2020), 106200. https://doi.org/10.1016/j.infsof.2019.106200

[71] Clemens Linnhoff, Philipp Rosenberger, Simon Schmidt, Lukas Elster, Rainer Stark, and Hermann Winner. 2021. Towards Serious Perception Sensor Simulation for Safety Validation of Automated Driving - A Collaborative Method to Specify Sensor Models. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, Indianapolis, IN, USA, 2688–2695. https://doi.org/10.1109/itsc48978.2021.9564661

[72] Angelica F. Magnussen, Nathan Le, Linghuan Hu, and W. Eric Wong. 2020. A Survey of the Inadequacies in Traffic Sign Recognition Systems for Autonomous Vehicles. *International Journal of Performability Engineering* 16, 10 (2020), 1588. https://doi.org/10.23940/ijpe.20.10.p10.15881597

[73] Helmut Martin, Bernhard Winkler, Stephanie Grubmuller, and Daniel Watzenig. 2019. Identification of performance limitations of sensing technologies for automated driving.

In *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*. IEEE, Graz, Austria, 1–6. https://doi.org/10.1109/ICCVE45908.2019.8965181

[74] Till Menzel, Gerrit Bagschik, and Markus Maurer. 2018. Scenarios for Development, Test and Validation of Automated Vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Changshu, China. https://doi.org/10.1109/IVS.2018.8500406

[75] Max-Arno Meyer, Mehdi Zouari, Sebastian Bannenberg, Markus Deppe, Sébastien Christiaens, Sung-Yong Lee, and Jakob Andert. 2025. Machine-readable specification and intelligent cloud-based execution of logical test cases for automated driving functions. *Automated Software Engineering* 32, 1 (2025), 1–30. https://doi.org/10.1007/s10515-024-00481-6

[76] Michael Wayland (CNBC). 2025. Why 2025 is set to be a crucial year for Amazon's Zoox robotaxi unit. https://www.cnbc.com/2025/01/17/amazon-zoox-plans-commercial-expansion.html (last accessed on 12.05.2025).

[77] Francesco Montanari, Christoph Stadler, Jorg Sichermann, Reinhard German, and Anatoli Djanatliev. 2021. Maneuver-based Resimulation of Driving Scenarios based on Real Driving Data. In *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Nagoya, Japan. https://doi.org/10.1109/IV48863.2021.9575441

[78] Gaurang Naik and Jung-Min Jerry Park. 2019. Impact of Wi-Fi Transmissions on C-V2X Performance. In *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, Newark, NJ, USA, 1–10. https://doi.org/10.1109/DySPAN.2019.8935647

[79] Hideaki Nanba, Yukihito Ikami, Kenichiro Imai, Kenji Kobayashi, and Manabu Sawada. 2018. An Advantage of the Vehicle to Vehicle Communication for an Automated Driving Car at the Encounter with an Ambulance. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E101.A, 9 (2018), 1281–1289. https://doi.org/10.1587/transfun.E101.A.1281

[80] Seo-Wook Park, Kunal Patil, Will Wilson, Mark Corless, Gabriel Choi, and Paul Adam. 2020. Creating Driving Scenarios from Recorded Vehicle Data for Validating Lane Centering System in Highway Traffic. In *SAE Technical Paper Series (ANNUAL)*. SAE International. https://doi.org/10.4271/2020-01-0718

[81] Liang Peng, Hong Wang, and Jun Li. 2021. Uncertainty Evaluation of Object Detection Algorithms for Autonomous Vehicles. *Automotive Innovation* 4, 3 (2021), 241–252. https://doi.org/10.1007/s42154-021-00154-0

[82] Robin Philipp, Jana Rehbein, Felix Grün, Lukas Hartjen, Zhijing Zhu, Fabian Schuldt, and Falk Howar. 2022. Systematization of Relevant Road Users for the Evaluation of Autonomous Vehicle Perception. In *2022 IEEE International Systems Conference (SysCon)*. IEEE, Montreal, Canada, 1–8. https://doi.org/10.1109/SysCon53536.2022.9773877

[83] Robin Philipp, Zhijing Zhu, Julian Fuchs, Lukas Hartjen, Fabian Schuldt, and Falk Howar. 2021. Automated 3D Object Reference Generation for the Evaluation of Autonomous Vehicle Perception. In *2021 5th International Conference on System Reliability and Safety (ICSRS)*. IEEE Computer Society, Palermo, Italy. https://doi.org/10.1109/ICSRS53853.2021.9660660

[84] Christopher Plachetka, Niels Maier, Jenny Fricke, Jan-Aike Termohlen, and Tim Fingscheidt. 2020. Terminology and Analysis of Map Deviations in Urban Domains: Towards Dependability for HD Maps in Automated Vehicles. In *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Las Vegas, NV, USA, 63–70. https://doi.org/10.1109/IV47402.2020.9304580

[85] Dariusz Pleban. 2013. Method of Testing of Sound Absorption Properties of Materials Intended for Ultrasonic Noise Protection. *Archives of Acoustics* 38, 2 (2013), 191–195. https://doi.org/10.2478/aoa-2013-0022

[86] D. Pomerleau and T. Jochem. 1996. Rapidly adapting machine vision for automated vehicle steering. *IEEE Expert* 11, 2 (1996), 19–27. https://doi.org/10.1109/64.491277

[87] Thomas Ponn, Matthias Breitfuß, Xiao Yu, and Frank Diermeyer. 2020. Identification of Challenging Highway-Scenarios for the Safety Validation of Automated Vehicles Based on Real Driving Data. In *2020 Fifteenth International Conference on Ecological Vehicles and Renewable Energies (EVER)*. IEEE, Monte-Carlo, Monaco, 1–10. https://doi.org/10.1109/EVER48776.2020.9242539

[88] Zhao Qidong, Zheng Tong, Zhang Yunshuang, Chen Chao, Zhang Qingyu, Zhao Shuai, and Du Zhibin. 2022. The Research on the Identification of ACC SOTIF Triggering Conditions Based on Scenario Analysis. In *2022 IEEE International Conference on Real-time Computing and Robotics (RCAR)*. IEEE, Guiyang, China, 263–266. https://doi.org/10.1109/RCAR54675.2022.9872207

[89] Rajesh Rajamani, Han-Shue Tan, Boon Kait Law, and Wei-Bin Zhang. 2000. Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons. *IEEE Transactions on Control Systems Technology* 8, 4 (2000), 695–708. https://doi.org/10.1109/87.852914

[90] Paul Rau, Christopher Becker, and John Brewer. 2019. Approach for deriving scenarios for safety of the intended functionality. In *26th International Technical Conference on the Enhanced Safety of Vehicles (ESV): Technology: Enabling a Safer Tomorrow. National Highway Traffic Safety Administration*. Eindhoven , Netherlands. https://www-esv.nhtsa.dot.gov/Proceedings/26/26ESV-000258.pdf

[91] Andreas Rausch and Manfred Broy. 2008. *Die V-Modell XT Grundlagen*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1–27. https://doi.org/10.1007/978-3-540-30250-6_1

[92] Lei Ren, Huilin Yin, Wancheng Ge, and Qian Meng. 2019. Environment Influences on Uncertainty of Object Detection for Automated Driving Systems. In *2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. IEEE, Suzhou, China, 1–5. https://doi.org/10.1109/CISP-BMEI48845.2019.8965948

[93] Elias Rocklage, Heiko Kraft, Abdullah Karatas, and Jörg Seewig. 2017. Automated scenario generation for regression testing of autonomous vehicles. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, Yokohama, Japan, 476–483. https://doi.org/10.1109/ITSC.2017.8317919

[94] Chang-Gyun Roh, Jisoo Kim, and I-Jeong Im. 2020. Analysis of Impact of Rain Conditions on ADAS. *Sensors* 20, 23 (2020). https://doi.org/10.3390/s20236720

[95] Kasih Omega Saerang. 2022. Syntax Types of Important Sentence Rules in English Writing. (2022). https://www.academia.edu/download/83413913/JOURNAL_SYNTAX_TYPES_OF_IMPORTANT_SENTENCE_RULES_IN_ENGLISH_WRITING.pdf

[96] Jan Sauerbier, Julian Bock, Hendrik Weber, and Lutz Eckstein. 2018. Definition of Scenarios for Safety Validation of Automated Driving Functions. *ATZ worldwide* 121, 1 (2018), 42–45. https://doi.org/10.1007/s38311-018-0197-2

[97] Maike Scholtes, Lukas Westhofen, Lara Ruth Turner, Katrin Lotto, Michael Schuldes, Hendrik Weber, Nicolas Wagener, Christian Neurohr, Martin Herbert Bollmann, Franziska Kortke, Johannes Hiller, Michael Hoss, Julian Bock, and Lutz Eckstein. 2021. 6-Layer Model for a Structured Description and Categorization of Urban Traffic and Environment. *IEEE Access* 9 (2021), 59131–59147. https://doi.org/10.1109/ACCESS.2021.3072739

[98] Fabian Schuldt. 2017. *Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen.* Ph. D. Dissertation. Technische Universität Braunschweig.

[99] Matthew Schwall, Tom Daniel, Trent Victor, Francesca Favaro, and Henning Hohnhold. 2020. Waymo Public Road Safety Performance Data. arXiv:2011.00038 [cs.RO] https://arxiv.org/abs/2011.00038

[100] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. 2018. On a Formal Model of Safe and Scalable Self-driving Cars. arXiv:1708.06374v6 [cs.RO, cs.AI, stat.ML] http://arxiv.org/abs/1708.06374v6

[101] Omveer Sharma, N.C. Sahoo, and N.B. Puhan. 2021. Recent advances in motion and behavior planning techniques for software architecture of autonomous vehicles: A state-of-the-art survey. *Engineering Applications of Artificial Intelligence* 101 (2021), 104211. https://doi.org/10.1016/j.engappai.2021.104211

[102] Jan Erik Stellet, Tino Brade, Alexander Poddey, Stefan Jesenski, and Wolfgang Branz. 2019. Formalisation and algorithmic approach to the automated driving validation problem. In *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Paris, France, 45–51. https://doi.org/10.1109/IVS.2019.8813894

[103] Leonard Stepien, Silvia Thal, Roman Henze, Hiroki Nakamura, Jacobo Antona-Makoshi, Nobuyuki Uchida, and Pongsathorn Raksincharoensak. 2021. Applying Heuristics to Generate Test Cases for Automated Driving Safety Evaluation. *Applied Sciences* 11, 21 (2021). https://doi.org/10.3390/app112110166

[104] Torben Stolte, Gerrit Bagschik, and Markus Maurer. 2016. Safety goals and functional safety requirements for actuation systems of automated vehicles. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, Rio de Janeiro, Brazil, 2191–2198. https://doi.org/10.1109/ITSC.2016.7795910

[105] Omer Sahin Tas and Christoph Stiller. 2018. Limited Visibility and Uncertainty Aware Motion Planning for Automated Driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Changshu, China, 1171–1178. https://doi.org/10.1109/IVS.2018.8500369

[106] Silvia Thal, Philip Wallis, Roman Henze, Ryo Hasegawa, Hiroki Nakamura, Sou Kitajima, and Genya Abe. 2023. Towards Realistic, Safety-Critical and Complete Test Case Catalogs for Safe Automated Driving in Urban Scenarios. In *2023 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Anchorage, AK, USA, 1–8. https://doi.org/10.1109/IV55152.2023.10186595

[107] Eric Thorn, Shawn C Kimmel, and Michelle Chaka. 2018. *A Framework for Automated Driving System Testable Cases and Scenarios (Report No. DOT HS 812 623)*. Technical Report. United States. Department of Transportation. National Highway Traffic Safety Administration, DOT HS 812 623. Washington, DC. https://rosap.ntl.bts.gov/view/dot/38824

[108] Jia Tong, Xingyu Xing, Runqing Guo, Wei Jiang, Lu Xiong, and Junyi Chen. 2022. Performance Limitations Analysis of Visual Sensors in Low Light Conditions Based on Field Test. In *SAE Technical Paper Series (ICVS 2022, Vol. 1)*. SAE International. https://doi.org/10.4271/2022-01-7086

[109] Sever Topan, Karen Leung, Yuxiao Chen, Pritish Tupekar, Edward Schmerling, Jonas Nilsson, Michael Cox, and Marco Pavone. 2022. Interaction-Dynamics-Aware Perception Zones for Obstacle Detection Safety Evaluation. In *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Aachen, Germany. https://doi.org/10.1109/IV51971.2022.9827409

[110] Jason Torchinsky. 2022. This Optical Illusion Of A Yawning Chasm In A Tunnel Messes With A Driver's Head Just Like It'll Mess With Yours. https://www.theautopian.com/this-optical-illusion-of-a-yawning-chasm-in-a-tunnel-ramp-messes-with-a-drivers-head-just-like-itll-mess-with-yours/ (last accessed on 11.05.2025).

[111] M.A. Turk, D.G. Morgenthaler, K.D. Gremban, and M. Marra. 1988. VITS-a vision system for autonomous land vehicle navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10, 3 (1988), 342–361. https://doi.org/10.1109/34.3899

[112] UK Parliament. 2024. Automated Vehicles Act 2024. https://www.legislation.gov.uk/ukpga/2024/10 (last accessed on 01.05.2025).

[113] UL Standards & Engagement. 2022. *ANSI/UL 4600 Standard for Safety for the Evaluation of Autonomous Products*. Standard. https://ulse.org/ul-standards-engagement/autonomous-vehicle-technology (last accessed on 17.12.2024).

[114] Simon Ulbrich, Till Menzel, Andreas Reschka, Fabian Schuldt, and Markus Maurer. 2015. Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, Gran Canaria, Spain, 982–988. https://doi.org/10.1109/itsc.2015.164

[115] UNECE. 2025. Automated Driving System ADS. https://wiki.unece.org/pages/viewpage.action?pageId=238223362 (last accessed on 12.09.2025).

[116] United Nations Economic Commission for Europe (UNECE). 2025. UN Regulation No. 157 - Uniform provisions concerning the approval of vehicles with regard to Automated Lane Keeping Systems (E/ECE/TRANS/505/Rev.3/Add.156/Rev.1). https://unece.org/transport/vehicle-regulations-wp29/standards/addenda-1958-agreement-regulations-141-160 (last accessed on 01.05.2025).

[117] Jorge Vargas, Suleiman Alsweiss, Onur Toker, Rahul Razdan, and Joshua Santos. 2021. An Overview of Autonomous Vehicles Sensors and Their Vulnerability to Weather Conditions. *Sensors* 21, 16 (2021), 5397. https://doi.org/10.3390/s21165397

[118] Walther Wachenfeld, Philipp Junietz, Raphael Wenzel, and Hermann Winner. 2016. The worst-time-to-collision metric for situation identification. In *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Gothenburg, Sweden, 729–734. https://doi.org/10.1109/IVS.2016.7535468

[119] Walther Wachenfeld and Hermann Winner. 2016. The Release of Autonomous Vehicles. In *Autonomous Driving*. Springer Berlin Heidelberg, 425–449. https://doi.org/10.1007/978-3-662-48847-8_21

[120] Sebastian Wagner, Korbinian Groh, Thomas Kuhbeck, Michael Dorfel, and Alois Knoll. 2018. Using Time-to-React based on Naturalistic Traffic Object Behavior for Scenario-Based Risk Assessment of Automated Driving. In *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Changshu, China, 1521–1528. https://doi.org/10.1109/IVS.2018.8500624

[121] Song Wang and Zhixia Li. 2019. Exploring causes and effects of automated vehicle disengagement using statistical modeling and classification tree based on field test

data. *Accident Analysis & Prevention* 129 (2019), 44–54. https://doi.org/10.1016/j.aap.2019.04.015

[122] Andrzej Wardziński. 2008. *Safety Assurance Strategies for Autonomous Vehicles*. Springer Berlin Heidelberg, 277–290. https://doi.org/10.1007/978-3-540-87698-4_24

[123] Wiktionary. [n. d.]. 抛磚引玉. https://en.wiktionary.org/wiki/%E6%8B%8B%E7%A3%9A%E5%BC%95%E7%8E%89 last accessed on 19.02.2025.

[124] Xingyu Xing, Tong Jia, Junyi Chen, Lu Xiong, and Zhuoping Yu. 2022. An Ontology-based Method to Identify Triggering Conditions for Perception Insufficiency of Autonomous Vehicles. arXiv:2210.08724 [eess.IV] https://arxiv.org/abs/2210.08724

[125] Zhang Xinxin, Li Fei, and Wu Xiangbin. 2020. CSG: Critical Scenario Generation from Real Traffic Accidents, In 2020 IEEE Intelligent Vehicles Symposium (IV). *2020 IEEE Intelligent Vehicles Symposium (IV)*, 1330–1336. https://doi.org/10.1109/iv47402.2020.9304609

[126] Yuki Yoshihara, Yoichi Morales, Naoki Akai, Eijiro Takeuchi, and Yoshiki Ninomiya. 2017. Autonomous predictive driving for blind intersections. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Vancouver, BC, Canada. https://doi.org/10.1109/IROS.2017.8206185

[127] Xudong Zhang and Yan Cai. 2023. Building Critical Testing Scenarios for Autonomous Driving from Real Accidents. In *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA '23)*. ACM, New York, NY, United States, 462–474. https://doi.org/10.1145/3597926.3598070

[128] Xinhai Zhang, Jianbo Tao, Kaige Tan, Martin Törngren, José Manuel Gaspar Sánchez, Muhammad Rusyadi Ramli, Xin Tao, Magnus Gyllenhammar, Franz Wotawa, Naveen Mohan, Mihai Nica, and Hermann Felbinger. 2023. Finding Critical Scenarios for Automated Driving Systems: A Systematic Mapping Study. *IEEE Transactions on Software Engineering* 49, 3 (2023), 991–1026. https://doi.org/10.1109/TSE.2022.3170122

[129] Ding Zhao, Henry Lam, Huei Peng, Shan Bao, David J. LeBlanc, Kazutoshi Nobukawa, and Christopher S. Pan. 2017. Accelerated Evaluation of Automated Vehicles Safety in Lane-Change Scenarios Based on Importance Sampling Techniques. *IEEE Transactions on Intelligent Transportation Systems* 18, 3 (2017), 595–607. https://doi.org/10.1109/TITS.2016.2582208

[130] Yuan Zhou, Yang Sun, Yun Tang, Yuqi Chen, Jun Sun, Christopher M. Poskitt, Yang Liu, and Zijiang Yang. 2023. Specification-Based Autonomous Driving System Testing. *IEEE Transactions on Software Engineering* 49, 6 (2023), 3391–3410. https://doi.org/10.1109/TSE.2023.3254142

[131] Zhijing Zhu, Robin Philipp, and Falk Howar. 2025. Leveraging Triggering Conditions for Efficient Scenario-Based Testing of Automated Vehicles. *SAE International Journal*

of Connected and Automated Vehicles 8, 4 (2025). https://doi.org/10.4271/12-08-04-0035

[132] Zhijing Zhu, Robin Philipp, Constanze Hungar, and Falk Howar. 2022. Systematization and Identification of Triggering Conditions: A Preliminary Step for Efficient Testing of Autonomous Vehicles. In *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Aachen, Germany, 798–805. https://doi.org/10.1109/IV51971.2022.9827238

[133] Zhijing Zhu, Robin Philipp, Constanze Hungar, and Falk Howar. 2024. Identifying Difficult Environmental Conditions with Scenario-Based Hazard and Fault Analysis. In *Computer Safety, Reliability, and Security. SAFECOMP 2024 Workshops*. Springer Nature Switzerland, 134–147. https://doi.org/10.1007/978-3-031-68738-9_10

[134] Zhijing Zhu, Robin Philipp, Yongqi Zhao, Constanze Hungar, Jürgen Pannek, and Falk Howar. 2023. Automatic Disengagement Scenario Reconstruction Based on Urban Test Drives of Automated Vehicles. In *2023 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Anchorage, AK, USA, 1–8. https://doi.org/10.1109/IV55152.2023.10186640

[135] Marc René Zofka, Florian Kuhnt, Ralf Kohlhaas, Christoph Rist, Thomas Schamm, and J. Marius Zöllner. 2015. Data-driven simulation and parametrization of traffic scenarios for the development of advanced driver assistance systems. In *2015 18th International Conference on Information Fusion (Fusion)*. Washington, DC, USA, 1422–1428.

Appendices

A. Catalog of Difficult Environmental Conditions

This chapter contains the full catalog of difficult environmental conditions, that is discussed in Chapter 6. The catalog has been previously published as supplementary material[1] for Paper III [133].

[1] https://doi.org/10.5281/zenodo.11472687

Table A.1.: Difficult environmental condition catalog (ROW = right of way, VRU = vulnerable road user, obj. = object, behav. = behavior, sit. = situation, traj. = trajectory, IA = Information Access, IR = Information Reception, IP = Information Processing, P = Plan, A = Action)

Cluster			Difficult Environmental Condition								
			EF1	ATTR1	BHV	INTR	EF2	ATTR2	TSC	SL	Functional Insufficiencies
Ambiguity of traffic guidance	Missing	I	Lane marks	missing					on turn lane	3	map/reality discrepancy (P)
	Aging / Damaged	I	Lane marks	faded						3	lane mark detection (IP)
		I	Lane marks	faded					on turn lane	3	lane mark detection (IP)
		I	Solid lane mark	faded						3	lane mark detection (IP)
		I	Zebra crossing	faded						3	road mark detection (IP)
		I	Road marks	faded						3	road mark detection (IP)
		I	Lane marks	crumbled						3	lane mark detection (IP)
	Temporary Modification	I	Negation	missing					on lane marks	3	sit. understanding, map/reality discrepancy (P)
		I	Negation	missing					on traffic sign	3	sit. understanding, map/reality discrepancy (P)
		I	Negation	unusual					on temp. traffic sign	3	traffic sign detection (IP)
		I	Yellow lane marks	inconsistent						3	lane mark detection (IP)
		I	Traffic light	deactivated						3	traffic light detection (IP), map/reality discrepancy (P)
		I	Temp. traffic sign							3	map/reality discrepancy (P)
		I	Yellow lane mark	remaining						3	map/reality discrepancy (P)
		I	Temp. traffic sign	remaining						3	map/reality discrepancy (P)
		III	Construction			partially blocks	ego lane			3	traj. generation (P)
		I	Roadside mod.	unusual						3	obj. behav. prediction (P)
		I	Road geometry mod.							3	traj. generation (P)
	Unregistered	I	Construction site	unregistered						3	map correctness (IA)
		I	Traffic guidance obj.	unregistered						1/3	map correctness (IA)
		I	Road marking	unregistered						1/3	map correctness (IA)
		I	Road geometry	unregistered						1	map correctness (IA)

Cluster			EF1	ATTR1	BHV	INTR	EF2	ATTR2	TSC	SL	Functional Insufficiencies
Road User behavior	Front Road User behavior	II	Cyclist	leading	leaves				its lane	4	obj. detection (IP), sit. understanding (P)
		II	Cyclist	leading	drives				on edge of its lane	4	obj. detection (IP), lane matching (P)
		II	Cyclist	leading	stops				on edge of its lane	4	obj. behav. prediction, traj. generation (P)
		II	Truck	leading	deploys loading dock				in ego lane	4	obj. detection & classification (IP)
		II	Vehicle	leading	waits				behind traffic jam	4	radar-based obj. detection (IR)
		II	Vehicle	leading	stops with hazard flasher					4	obj. behav. prediction, sit. understanding (P)
		II	Vehicle	leading	drives				on lane markings	4	obj. behav. prediction, lane matching (P)
		I	Traffic jam	leading					next to free lane	4	obj. behav. prediction, traj. generation (P)
	Side Road User behavior	II	Vehicle	without ROW	brakes too late				from merging lane	4	obj. behav. prediction, traj. generation (P)
		II	Pedestrian		stops abruptly				in front of crosswalk	4	obj. behav. prediction, traj. generation (P)
		II	Multiple vehicles		cut in				to ego lane	4	obj. behav. prediction, traj. generation (P)
		I	Multiple vehicles	parking					along road side	4	obj. segmentation (IP)
		II	Truck	with trailer	cuts into				ego lane	4	obj. behav. prediction, traj. generation (P)
		II	Truck	with trailer	cuts in without turn indication				to ego lane	4	obj. behav. prediction, traj. generation (P)
		I	Vehicle	parking					on the lane	4	obj. behav. prediction, traj. generation (P)
		II	Cyclist		crosses				from roadside	4	field of view (IR), obj. behav. prediction (P)
		II	VRU		jaywalks					4	obj. behav. prediction, traj. generation (P)
		II	Vehicle		stops				on negated bus lane	4	map/reality discrepancy, obj. behav. predicition (P)
		III	Vehicles	parking		narrow	road		in both directions	4	obj. behav. predicition, traj. generation (P)
		II	Animal		crosses abruptly					4	obj. behav. predicition, traj. generation (P)

Cluster		EF1	ATTR1	BHV	INTR	EF2	ATTR2	TSC	SL	Functional Insufficiencies
Opposite Road User behavior	II	Vehicle	oncoming, in dark color	drives				under tree shadows	4,2	camera - obj. detection (IP)
	III	Vehicle	oncoming		overtakes	vehicle	standing	over ego lane	4	obj. behav. predicition, traj. generation (P)
	II	E-scooter	oncoming	drives				under tree shadows	4,2	camera - obj. detection (IP)
	II	Cyclist	oncoming	drives				on the wrong side	4	obj. behav. predicition, traj. generation (P)
	I	Traffic jam						on opposite lane	4	obj. behav. prediction (P)
Object Proximity in Space	I	Traffic lights	in close proximity						1	traffic light detection (IP)
	I	Grass						around roadside barrier	2	object classification (IP)
	III	Workers			unload	truck		on road	4	obj. detection (IP), obj. behav. predicition (P)
	III	Workers			shut	truck rear door		on road	4	obj. detection (IP), obj. behav. predicition (P)
	I	Traffic signs	in close proximity						1	traffic sign detection (IP)
	I	Movable objects	in close proximity						4	object segmentation (IP)
	I	VRU						next to barrier	4,2	object segmentation (IP)
	I	Pedestrian						next to tree	4,2	object segmentation (IP)
	II	Vehicle	tall	drives				under tree branches	4,2	object segmentation (IP)
Unusual Positioning	I	Traffic light						at unusual position	1	traffic light matching (IP)
	II	E-Scooter		drives				on motor lane	4	sit. understanding, obj. behav. prediction (P)
Unusual Pose	I	Cyclist	in unusual pose						4	object detection (IP)
	I	Traffic sign	bent						3	traffic sign detection (IP)
	I	Recumbent bicycle							4	object detection (IP)

Features / Status of Objects	Cluster	EF1	ATTR1	BHV	INTR	EF2	ATTR2	TSC	SL	Functional Insufficiencies
Object Appearance	I	Vehicle	w/ traffic light sticker						4	object detection, traffic light detection (IP)
	I	Truck	with open rear					in ego lane	4	object detection (IP)
	I	Static object	pole-alike					on roadside	2	land mark detection (IP)
	I	Vehicle	w/ red-white-stripes						4	object classification (IP)
	I	Object	w/ thin-profile						1-4	radar RCS (IR)
	I	Object	w/ low RCS						1-4	Radar sensor performance (IR)
	I	Human photo						on rear of trailer	4	camera - object classification (IP)
	I	Truck	w/o flat rear						4	object detection (IP)
	I	Vehicle	w/ low contrast to background						4	camera - object detection (IP)
	I	Vehicle	leading, in ground color						4	camera - object detection (IP)
	I	Vehicle	oncoming, in ground color						4	camera - object detection (IP)
	I	Median curb/strip	in road color						1	lane detection (IP)
	I	Lane marks	w/ low contrast to road						1	lane mark detection (IP)
	I	Bicycle lane	in red						1	lane detection (IP)
Unusual Appearance	I	Cyclist	in unusual size						4	object detection/classification (IP)
	I	Tandem bicycle							4	object classification (IP)
	I	Vehicle	with open door					on roadside	4	object detection (IP), traj. generation (P)
	I	Vehicle	w/ unusual shape						4	object classification (IP)
	I	Cyclist	w/ unusual shape						4	object classification (IP)
Reflectivity	I	Goods vehicle	w/ reflective rear						4	lidar / camera (IR)
	I	Object	w/ low reflection					on road	3/4	lidar / camera (IR), object detection (IP)
	I	Object	w/ high reflection					on road	3/4	lidar / camera (IR), object detection (IP)

Cluster			EF1	ATTR1	BHV	INTR	EF2	ATTR2	TSC	SL	Functional Insufficiencies
			Junction	w/ prescr. directions						1	map correctness (IA)
			Bus lane	shared, marked						1	situational understanding (P)
		Road Geometry	Road	w/ high curvature						1	map correctness (IA), steering (A)
			Road	w/ high slope (down)						1	map correctness (IA), braking/accelerating (A)
			Road	w/ high slope (up)						1	information access (IA)
		Road Surface Condition	Road bumps							1/3	sensor calibration, information access (IA)
			Road	slippery						5	braking torque setting (A)
			Road	wet						5	lidar (IR), braking torque setting (A)
			Lane	w/ tire markings					on intersection area	3	lane detection (IP)
			Lane	w/ wet leafs						3	lane detection (IP), braking / accelerating (A)
III	**Road Condition**		Dirt			covers	ego lane			3	lane detection (IP), sit. understanding (P)
III			Snow			covers	ego lane			3	lane detection (IP), sit. understanding (P)
			Rail tracks						on ego lane	1	sit. understanding (P), braking / accelerating / steering (A)
			Small object						in ego lane	4	obj. behav. prediction, traj. generation (P)
		Fixed Roadside Structure	Tree branch						above road	2	situational understanding (P)
			Tree	newly planted						2	map info missing (IA), map/reality discrepancy (P)
			Tall buildings						on roadsides	2	GPS signal (IA)
			Trees	lined					on rural roadside	2	land mark detection (IP)
			Tree (land mark)	leafless					in autumn	2	land mark detection (IP)
			Land mark	with changed pose						3	land mark detection (IP)

Cluster			EF1	ATTR1	BHV	INTR	EF2	ATTR2	TSC	SL	Functional Insufficiencies
V2X Connectivity		I	Mobile network	w/ high load						6	remote guidance (IA)
		I	Mobile network	unstable						6	remote guidance (IA)
		I	Road user	w/o V2X service						4,6	Car2X availability (IA)
		II	V2X infrastructure		sends incorrect info					6	map/reality discrepancy (P)
		II	V2X infrastructure		sends incorrect road friction estimation					6	map/reality discrepancy (P)
Atmosphere	Weather Condition	I	Weather	adverse						5	Sensor performance (IR)
		I	Rain	heavy						5	Sensor performance (IR)
		I	Snow	heavy						5	Sensor performance (IR)
		I	Wind	heavy						5	steering (A)
		I	Fog							5	Sensor performance (IR)
		I	Rain	light						5	Sensor performance (IR)
		I	Snow	light						5	Sensor performance (IR)
		I	Gusts	short, strong						5	steering (A)
	Optical phenomena	III	Barrier			throws	shadow	in stripe shape	on lane	2,5	lane mark detection (IP)
		III	Sunlight			reflects on	oncoming vehicle		on narrow lane	5,4	obj. detection (IP)
		III	Sunlight			reflects on	road surface			5	lane mark detection, obj. detection (IP)
		III	Street light			reflects on	road surface	wet		5	lane mark detection, obj. detection (IP)
		III	Sunlight			throws	shadow	in stripe shape	on lane	5	lane mark detection (IP)
		I	Sun glare							5	camera performance (IR)
	Illumination	III	Street lights	sparsely distributed		poorly illuminate	road		at night	5	obj. / lane mark detection (IP)
		I	Illumination	adverse						5	obj. / lane mark detection (IP)
		I	Illumination	spatially altering						5	obj. / lane mark detection (IP)
	Particulates	I	Vehicle emission							5	obj. detection (IP)

Cluster		EF1	ATTR1	BHV	INTR	EF2	ATTR2	TSC	SL	Functional Insufficiencies
Occlusion	I	Stop sign	initially occluded						1	traffic sign detection (IP)
	I	ROW sign	fully occluded						1	traffic sign detection (IP), map/reality discrepancy (P)
	I	ROW sign	partially occluded						1	traffic sign detection (IP)
	III	Leafs			cover	road boundaries			3	lane boundaries detection (IP)
	III	Grass			covers	curb			3	lane boundaries detection (IP)
	III	Leafs			cover	lane markings			3	lane mark detection (IP)
	III	Dirt			covers	lane markings			3	lane mark detection (IP)
	III	Stickers			partially cover	ROW sign			3	traffic sign detection (IP)
	III	Stickers			partially cover	traffic sign			3	traffic sign detection (IP)
	III	Snow			covers	traffic sign			3	traffic sign detection (IP), map/reality discrepancy (P)
	III	Snow			covers	lane markings			3	lane mark detection (IP)
	III	Snow			covers	curb			3	lane boundaries detection (IP)
	III	Building			occludes	road user			2,4	obj. detection (IP)
	III	Dirt			covers	yellow lane markings			3	lane mark detection (IP)
	III	Stickers			cover	delineator			3	lane boundaries detection (IP)
	III	Roadworks barrier			partially occludes	cyclist	crossing		3,4	obj. detection (IP)
	III	Roadworks barrier			partially occludes	pedestrian	crossing		3,4	obj. detection (IP)
	III	Roadworks barrier			partially occludes	vehicle	merging	in ego lane	3,4	obj. detection (IP)
	III	Vehicles	parking		occlude	pedestrians			4	obj. detection (IP)
	III	Snow			covers	yellow lane markings			3	lane mark detection (IP)