# Functional Decomposition Approach - Reducing the Safety Validation Effort for Highly Automated Driving

Vom Fachbereich Maschinenbau an der
Technischen Universität Darmstadt
zur Erlangung des Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte

# Dissertation

vorgelegt von

**Christian Thomas Amersbach, M. Sc.**
aus Würzburg

| | |
|---|---|
| Berichterstatter: | Prof. Dr. rer. nat. Hermann Winner |
| Mitberichterstatter: | Prof. Dr.-Ing. Markus Maurer |
| Tag der Einreichung: | 26.09.2019 |
| Tag der mündlichen Prüfung: | 26.02.2020 |

Darmstadt 2020
D 17

# Preface

This dissertation was written as part of my work as a research assistant at the Institute of Automotive Engineering (FZD) at TU Darmstadt. During this time, I could contribute to the projects PEGASUS and in part-time UNICAR*agil* as well as VVM. Therefore, I would like to thank the sponsors BMWi and BMBF, without whose funding these projects and this dissertation would not have been possible. In addition, I would also like to thank all the project staff who contributed to the development of this work with many valuable discussions.

Big thank you to all my colleagues at FZD - this also applies explicitly to the secretarial office and the workshop - for the phenomenal working atmosphere. I will probably miss that the most when I leave here. In addition, of course, I would like to thank all my colleagues who contributed to the development of this dissertation with their discussions and feedback. I would also like to thank all the students who supported me in my research work.

Special thanks go to my doctoral supervisor Prof. Dr. rer. nat. Hermann Winner for the trust he placed in me and, above all, for the excellent support and feedback during my time as a doctoral candidate. Additionally, I thank Prof. Dr.-Ing. Markus Maurer for taking over the role as second examiner.

I would also like to thank all my friends and colleagues who always had an open ear and time for a beer when things did not go so well or when there was something to celebrate.

Finally, but importantly, I would like to thank my family, who are always there for me.

Darmstadt, September 2019.

# List of Contents

**Preface** ..... **I**

**List of Contents** ..... **II**

**List of Abbreviations** ..... **V**

**List of Symbols and Indices** ..... **VII**

**List of Figures and Tables** ..... **VIII**

**Kurzzusammenfassung** ..... **X**

**Abstract** ..... **XI**

**1 Introduction** ..... **1**
1.1 Current Test Concepts and the Approval Trap for Highly Automated Vehicles ..... 1
1.2 Motivation and Research Theses ..... 4
1.3 Structure of the Dissertation ..... 6

**2 State of the Art** ..... **8**
2.1 Safety Verification and Validation of Highly Automated Driving ..... 8
2.1.1 Approaches for the Safety Verification and Validation of HAV ..... 8
2.1.2 Test Environments ..... 19
2.1.3 Evaluation Criteria and Safety Requirements ..... 22
2.1.4 Test Coverage ..... 24
2.1.5 Approaches to Reduce Validation Effort ..... 27
2.1.6 Summary of the State of the Art for the Safety Verification and Validation of HAV ..... 31
2.2 Functional Decomposition ..... 33
2.2.1 Decomposition in General ..... 33
2.2.2 Functional Decomposition of the Driving Task ..... 35
2.2.3 Summary of the State of the Art on Decomposition ..... 44

**3 Objective and Research Questions** ..... **45**
3.1 Functional Decomposition as Safety Validation Method for HAD ..... 45
3.1.1 Scope ..... 45
3.1.2 Methodology ..... 46
3.1.3 Intended Benefits ..... 47
3.1.4 Requirements for Functional Decomposition as Validation Method ..... 48
3.2 Derivation of Research Questions ..... 50

**4 Functional Layers and Interfaces** ..... **52**

4.1 Definition of Functional Layers ..... 52
4.1.1 Layer 0: Information Access ..... 53
4.1.2 Layer 1: Information Reception ..... 54
4.1.3 Layer 2: Information Processing ..... 54
4.1.4 Layer 3: Situational Understanding ..... 55
4.1.5 Layer 4: Behavioral Decision ..... 55
4.1.6 Layer 5: Action ..... 55
4.2 Definition of Generic and Observable Interfaces ..... 56
4.2.1 Input to Layer 0: Existing Information ..... 56
4.2.2 Interface between Layer 0 and Layer 1: Accessible Information ..... 56
4.2.3 Interface between Layer 1 and Layer 2: Sensor Raw Data ..... 57
4.2.4 Interface between Layer 2 and Layer 3: Subjective Scene Representation ..... 57
4.2.5 Interface between Layer 3 and Layer 4: Situation Representation ..... 58
4.2.6 Interface between Layer 4 and Layer 5: Planned Trajectory ..... 58
4.2.7 Output from Layer 5: Observable Behavior ..... 58
4.3 Summary and Exemplary Application of the Proposed Decomposition Scheme ..... 59
4.4 Interim Conclusion ..... 61

**5 Particulate Evaluation Criteria ..... 62**
5.1 Methodology to Derive Particulate Evaluation Criteria ..... 62
5.2 Exemplary Derivation of Evaluation Criteria Using FTA ..... 64
5.3 Exemplary Derivation of Evaluation Criteria Using STPA ..... 65
5.4 Automatic Derivation of Dependency Chains ..... 68
5.5 Interim Conclusion ..... 68

**6 Allocation of Influence Parameters to Functional Layers ..... 70**
6.1 Selection and Analysis of External Influence Parameters on System Level ..... 70
6.2 Allocation of External Influence Parameters to Functional Layers ..... 71
6.2.1 Intuitive Allocation of Influence Parameters ..... 72
6.2.2 Rule-Based Allocation of Influence Parameters ..... 72
6.3 Input Parameters on the Interfaces ..... 74
6.4 Interim Conclusion ..... 75

**7 Generation of Particulate Test Cases ..... 76**
7.1 Systematic Test Case Generation in General ..... 76
7.2 Systematic Generation of Particulate Test Cases ..... 78
7.3 Interim Conclusion ..... 80

**8 Quantification of the Potential to Reduce the Validation Effort ..... 81**
8.1 Parameter Space Explosion in Scenario-Based Validation ..... 81
8.1.1 Influence Parameters ..... 81
8.1.2 Combinatorics ..... 83
8.2 Estimating Required Test Coverage for Scenario-Based Validation ..... 84

8.3 Reduction Potential with Functional Decomposition ....86
8.3.1 Exemplary Scenario Set ....87
8.3.2 Particulate Parameter Spaces....88
8.3.3 Less Complex Subsystems Require a Smaller Test Coverage. ....89
8.3.4 Aggregation of Perception Layer Tests ....90
8.3.5 Total Reduction of the Test Suite Size by Functional Decomposition ....91
8.4 Comparing the Reduction Potential with Feasible Test Effort ....92
8.5 Boundary Conditions for Discretization of Influence Parameters....93
8.6 Interim Conclusion ....95

**9 Summary and Conclusion ....96**
9.1 Summary of the Results....96
9.2 Meeting the Objective and Requirements....98
9.3 Applicability in the Automotive Industry ....100
9.4 Remaining Research Questions ....101

**A Appendix ....102**
A.1 Exemplary Parameter Space ....102

**List of References ....104**

**Own Publications ....123**

**Supervised Theses ....124**

# List of Abbreviations

| Abbreviation | Description |
|---|---|
| *ACC* | adaptive cruise control |
| *ADAS* | advanced driving assistance system |
| *AEB* | autonomous emergency braking |
| *ALARP* | as low as reasonably practicable |
| *ASIL* | automotive safety integrity level |
| *CCA* | common cause analysis |
| *DAS* | driving assistance system |
| *DDT* | dynamic driving task |
| *DiL* | driver-in-the-loop |
| *EVT* | extreme value theory |
| *FMEA* | failure mode and effect analysis |
| *FTA* | fault tree analysis |
| *FTFI* | failure triggering fault interaction |
| *GAMAB* | French: “globalement au moins aussi bon”, generally at least as good as |
| *GM* | General Motors |
| *HAD* | highly automated driving |
| *HARA* | hazard analysis risk assessment |
| *HAV* | highly automated vehicle(s) |
| *HiL* | hardware-in-the-loop |
| *HMI* | human-machine-interface |
| *IPOG* | in-parameter-order-general |
| *ISO* | International Organization for Standardization |
| *lidar* | light detection and ranging |
| *MEM* | minimum endogenous mortality |
| *NCAP* | new car assessment program |
| *ODD* | operational design domain |
| *OEDR* | object and event detection and response |
| *OEM* | original equipment manufacturer |
| *OUT* | object under test |
| *PEGASUS* | **p**roject for the **e**stablishment of **g**enerally **a**ccepted quality criteria, tools and methods as well as **s**cenarios and **s**ituations for the release of highly-automated driving functions |
| *radar* | radio detection and ranging |
| *RCA* | root cause analysis |
| *RSS* | responsibility-sensitive safety |
| *SAE* | Society of Automotive Engineers |
| *SiL* | software-in-the-loop |
| *SPICE* | software process improvement and capability determination |

| Abbreviation | Description |
|---|---|
| *STPA* | system theoretic process analysis |
| *US* | United States (of America) |
| *V2X* | vehicle-to-everything |
| *VAAFO* | virtual assessment of automation in field operation |
| *VEHiL* | vehicle-hardware-in-the-loop |
| *ViL* | vehicle-in-the-loop |
| *XiL* | X-in-the-loop |

# List of Symbols and Indices

| Symbol | Unit | Description |
|---|---|---|
| $e$ | % | significance level |
| $f$ | ./. | factor |
| $k$ | ./. | number of instances per parameter |
| $N$ | ./. | total number of parameters |
| $P$ | % | probability |
| $p$ | ./. | parameter |
| $s$ | m, km | distance |
| $S$ | ./. | size of the test suite |
| $t$ | ./. | number of parameters considered for $t$-wise coverage |
| $\tau$ | s, weeks | time |
| $v$ | m/s | velocity |

| Index | Description |
|---|---|
| d | discretization |
| eff | effort |
| $i$ | $i$-th element |
| $N$ | $N$-wise |
| o | overlap |
| p | parallelization |
| part | particulate |
| ref | reference |
| req | required |
| rt | real-time |
| s | distance |
| sc | scenario |
| SP | safety performance |
| suc | success |
| sys | system |
| $t$ | $t$-wise |
| tot | total |
| u | unique |

# List of Figures and Tables

Figure 1-1: Illustration of the driving task taken over by HAV..... 4
Figure 1-2: Structure of the dissertation. ..... 7
Figure 2-1: Definition of relevant, complex and critical scenarios..... 12
Figure 2-2: 6-layer model for scenario description..... 13
Figure 2-3: Comparison of data-driven and knowledge-driven scenario creation..... 15
Figure 2-4: Overview of the PEGASUS method. ..... 17
Figure 2-5: Classification of test environments according to Wachenfeld and Winner. ..... 21
Figure 2-6: Classification of test environments according to Steimle et al.. ..... 21
Figure 2-7: Macroscopic safety requirements for different stakeholders. ..... 23
Figure 2-8: Fault rate for different FTFI numbers in different domains..... 26
Figure 2-9: Number of surprises per distance covered. ..... 33
Figure 2-10: Different viewpoints on system architectures. ..... 35
Figure 2-11: Categories of human target-oriented behavior and three-level hierarchy of driving task ..... 36
Figure 2-12: Endsley's model of situation awareness in dynamic decision-making. ..... 37
Figure 2-13: 4-layer model of information processing after Wickens. ..... 39
Figure 2-14: Model of internal malfunction by Rasmussen and Zimmer. ..... 39
Figure 2-15: Model of unsafe actions after Reason. ..... 40
Figure 2-16: Error classification after Hacker. ..... 40
Figure 2-17: Reference architecture for scalable cooperative automation by Lotz. ..... 42
Figure 2-18: Functional system architecture for an autonomous on-road vehicle according to Matthaei and Maurer. ..... 43
Figure 2-19: Simplified ACC ability graph. ..... 43
Figure 3-1: Methodology Overview. ..... 46
Figure 4-1: Functional layers and interfaces. ..... 59
Figure 4-2: Decomposition scheme applied to the system architecture by Lotz. ..... 60
Figure 4-3: Decomposition scheme applied to the system architecture by Matthaei. ..... 61
Figure 5-1: Methodology to derive particulate evaluation criteria. ..... 63
Figure 5-2: "Swiss scenario". ..... 64
Figure 5-3: Simplified exemplary failure tree. ..... 65
Figure 5-4: Control Circuit for STPA of a decomposed HAD function. ..... 66
Figure 5-5: Exemplary derivation of particulate evaluation criteria using STPA. ..... 67
Figure 5-6: Exemplary dependency chain. ..... 68
Figure 5-7: Exemplary task-chain pattern skill graph. ..... 69
Figure 6-1: Selection and analysis of influence parameters according to Schuldt. ..... 71
Figure 6-2: Decision tree for the allocation of influence parameters. ..... 73
Figure 6-3: Open-loop (left) and closed-loop (right) test harnesses according to Schuldt. ..... 74
Figure 6-4: Generation of input data for particulate tests. ..... 75
Figure 7-1: Procedure for a test concept according to Schuldt et al. ..... 76
Figure 7-2: Systematic test case generation. ..... 77
Figure 7-3: Allocation of test cases to XiL methods according to Schuldt. ..... 79
Figure 8-1: Exemplary scenario set. ..... 87
Figure 8-2: Size of *t*-wise test suites for exemplary scenarios. ..... 88

Figure 8-3: Reduction of the test suite size by parameter space reduction through functional decomposition for different *t*-wise test coverages. ..... 89
Figure 8-4: Reduction of the test suite size by aggregation of parameters in an equivalency class scenario. ..... 91
Figure 8-5: Potential reduction of the test suite size by functional decomposition. ..... 91
Figure 8-6: Correlation between $f_d$, $t$ and $f_{eff}$. ..... 94

Table 1-1: Summary of the levels of driving automation. ..... 3
Table 2-1: Variety of number of parameters and discretization steps of logical scenarios. ..... 14
Table 2-2: Overview of verification and validation approaches and test environments. ..... 32
Table 2-3: Summary of the comparison of action theoretic models for error classification by Gründl. ..... 38
Table 8-1: Assumed discretization for exemplary parameters. ..... 82

Table A - 1: Exemplary parameter space. ..... 102
Table A - 2: Sizes of the resulting test suites ..... 103

# Kurzzusammenfassung

Diese Dissertation befasst sich mit der Anwendung der aus anderen Bereichen, bspw. der Mathematik oder Informatik, bekannten Methode der Funktionalen Dekomposition zur Validierung automatisierter Fahrfunktionen. Diese ermöglicht, den nötigen Testaufwand im Vergleich zu Szenario-basierten Black-Box-Tests des Gesamtsystems zu reduzieren.

Im ersten Teil dieser Arbeit wird der Stand der Technik zur Verifikation und Validierung automatisierter Fahrfunktionen analysiert. Insbesondere aufgrund der sehr hohen Testumfänge, die durch die „offene Welt“ mit einer Vielzahl an Einflussparametern entstehen, ist eine Validierung von höher automatisierten Fahrfunktionen mit existierenden Methoden nicht realisierbar. Dies zeigt die Notwendigkeit, neue Ansätze zur Reduktion des Validierungsaufwandes zu entwickeln. Die Entwicklung einer Methode zur Anwendung der Funktionalen Dekomposition zur Definition von sog. Partikulären Testfällen, d.h. Testfällen, die der Verifizierung und schließlich Validierung einer oder mehrerer funktionalen Ebenen einer automatisierten Fahrfunktion dienen, wird deshalb als Ziel dieser Arbeit definiert. Anschließend werden Anforderungen an die Entwicklung einer Validierungsmethode definiert und Forschungsfragen abgeleitet.

Im Hauptteil der Arbeit werden diese Forschungsfragen bearbeitet und dabei die folgenden Teilschritte der Methode entwickelt:

- Definition unabhängiger funktionaler Ebenen und deren Schnittstellen
- Definition von Kriterien zur Bewertung der funktionalen Ebenen
- Zuordnung von Einflussparametern
- Erstellung Partikulärer Testfälle

Weiterhin wird das Potential der Funktionalen Dekomposition zur Reduktion des Testaufwandes quantifiziert. Hierbei zeigt sich, dass sowohl der absolute Testaufwand als auch die mögliche Reduktion des Testaufwandes mittels Funktionaler Dekomposition stark von der benötigten Testabdeckung sowie der Diskretisierung der Einflussparameter abhängen. Abhängig von der Testabdeckung kann die Anzahl der benötigten Testfälle mit der vorgestellten Methode um bis zu 2 Größenordnungen reduziert werden.

Im letzten Teil der Arbeit werden die zu Beginn gesteckten Ziele dem tatsächlichen Erkenntnisgewinn gegenübergestellt und verbleibende sowie neu hinzugekommene Fragestellungen für weitere Forschungsvorhaben aufgezeigt. Diese betreffen vor allem eine weitere Detailierung und Automatisierung der Methode für eine Anwendung in der Praxis, als auch weitere Maßnahmen zur Reduktion des Validierungsaufwandes über die Anwendung der Funktionalen Dekomposition hinaus wie bspw. die Definition der minimal nötigen Testraumabdeckung und die Diskretisierung von wertekontinuierlichen Einflussparametern.

# Abstract

This dissertation is concerned with the application of functional decomposition - which is known from other fields, for example mathematics or computer science - for the validation of automated driving functions. The approach aims to reduce the number of required test cases compared to a scenario-based black box system test.

The first part of this thesis analyzes the state of the art for the verification and validation of automated driving functions. A validation of highly automated driving functions with existing methods is not feasible due to the "open world" with a multitude of influence parameters that leads to a high number of required tests.

This challenge indicates the need to develop new approaches to reduce the validation effort. The development of a method for the application of functional decomposition for the definition of so-called particulate test cases - i.e. test cases that serve to verify and ultimately validate one or more functional layers of an automated driving function - is therefore defined as the goal of this work. Subsequently, requirements for the development of a validation method are defined and research questions are derived.

The main part of the thesis focuses on these research questions and the development of the following substeps of the methodology:

- Definition of independent functional layers and their interfaces
- Definition of criteria for evaluating the functional layers
- Allocation of influence parameters
- Generation of particulate test cases

Furthermore, the potential to reduce the validation effort by functional decomposition is quantified. This shows that the absolute test effort as well as the possible reduction of the test effort by functional decomposition strongly depend on the required test coverage and the discretization of the influence parameters. Depending on the required test coverage, the amount of required test cases can be reduced by up to two orders of magnitude by the introduced approach.

In the final part of this thesis, the goals set at the beginning are compared with the actual gain of knowledge and remaining as well as newly added questions for further research projects are presented. These concern above all a further detailing and automation of the method for an application in practice, as well as further measures for the reduction of the validation effort beyond the application of the functional decomposition as for example the definition of the minimum necessary test space coverage and the discretization of influence parameters with a continuous value range.

# 1 Introduction[1]

The technical development of autonomous vehicles has reached a level that would allow a market introduction soon. Highly automated vehicles (HAV, i.e. SAE level 3 and higher[2]) have been recently announced by different OEMs[3,4]. Prototypes of automated vehicles exist among OEMs, suppliers, and research facilities. The technology has been successfully demonstrated in public, e.g. with the autonomous Bertha Benz Drive by Mercedes[5], the project "Stadtpilot" of TU Braunschweig[6], and many more. However, except for demonstration vehicles and testing fleets[7] with limited operational radius in dedicated test fields[8], there are no HAV in operation in public traffic yet.

## 1.1 Current Test Concepts and the Approval Trap for Highly Automated Vehicles

One reason why the technology is still not available on the market are the high safety requirements for automated driving and the challenge to prove that such a system is safe enough. Firstly, the answer to the question "*How safe is safe enough?*" is still subject of various research projects and a holistically accepted answer has not yet been found. Secondly, once the safety requirements for HAV are specified and included in legislation, they have to be proven during the safety validation process.

For the approval of driver-only vehicles (i.e. SAE level 0[2]), it is assumed that all components are designed and approved according to industry standards such as ISO 26262[9] and therefore do not exceed maximum failure rates. Additionally, it is relied on the abilities of the driver to maneuver the vehicle reliably in traffic, which are proven with test drivers. Over the last

[1] Parts of this section have been published already in: Amersbach, C.; Winner, H.: Functional Decomposition to Reduce Approval Effort for HAD (2017).

[2] SAE: J3016: Taxonomy and Definitions for […] Vehicle Automated Driving Systems (2018).

[3] Audi MediaInfo: Automated driving at a new level: the Audi AI traffic jam pilot (2017).

[4] Sage, A.; Lienert, P.: GM plans large-scale launch of self-driving cars in U.S. cities in 2019 (2017).

[5] Ziegler, J. et al.: Making Bertha Drive—An Autonomous Journey on a Historic Route (2014).

[6] Wille, J. M. et al.: Stadtpilot: Driving autonomously on Braunschweig's inner ring road (2010).

[7] e.g.: Volkswagen AG: Volkswagen tests highly-automated driving in Hamburg (2019).

[8] e.g. in Germany: BMVI: Digital Test Beds (2019).

[9] ISO: ISO 26262: Road vehicles – Functional safety (2018).

decades, this has been shown to be a successful proof of safety. While already advanced driver assistance systems (ADAS; i.e. SAE level 1 and 2) take over some parts of the dynamic driving task (DDT), the driver is still responsible for object and event detection and response (OEDR) and acts as a permanent fallback level for the DDT (see Table 1-1). Thus, she/he has to supervise the system and its environment permanently and has to intervene if necessary. [10] Therefore, the *Code of Practice* [11] assumes that the responsibility for the vehicle's behavior still remains with the human driver who is always in the loop and can overwrite or deactivate the system at any time. This allows transferring the results of conducted tests with test drivers and studies with normal drivers (i.e. without special training) [12] to future users, similar to driver-only vehicles.[13a] However, as for HAV, the driver is not continuously in the loop anymore, the existing standards and the *Code of Practice* cannot be transferred to highly automated driving without adaptions. For example, when assigning automotive safety integrity levels (ASIL), the controllability of a situation by the driver has to be considered. However, as the driver is not responsible to monitor the system permanently, this controllability is not given.[14] If the driver is not responsible for the vehicle behavior at any time anymore, which is already the case for intervening emergency functions (operation mode C according to the classification of driver assistance systems and vehicle automation from Gasser et al. [15], e.g. automatic emergency braking), tests that only focus on the driver's controllability are not sufficient any more. Even if a lot of testing in the development phase is shifted to simulations, the final approval for any kind of driver assistance systems is still carried out in real driving tests. According to Christiansen[16], the approval of current level 2 systems requires up to 12 million test kilometers to validate that all safety, functional, quality and comfort requirements are met.

With the transition from assisted (i.e. SAE level $\leq 2$) to automated (i.e. SAE level $\geq 3$) driving, the entire DDT including navigation, guidance and stabilization/control and OEDR as well as its fallback level is taken over by the HAV (see Figure 1-1 and Table 1-1) and therefore has to be included in its validation process.[13b] This leads to new challenges for the validation and approval of HAV, the so-called "approval trap"[17, 13c].

---

[10] SAE: J3016: Taxonomy and Definitions for […] Vehicle Automated Driving Systems (2018).

[11] Cotter, S. et al.: The institutional context for ADAS: A code of practice for development (2006).

[12] Weitzel, D. A.: Diss., Kontrollierbarkeit nicht situationsgerechter Reaktionen von ADAS (2013), pp. 33 ff.

[13] Wachenfeld, W.; Winner, H.: The Release of Autonomous Vehicles (2016). a: pp. 428 ff.; b: p. 434; c: p. 437.

[14] Reschka, A. et al.: Ability and skill graphs (2015).

[15] Gasser, T. M. et al.: Framework Conditions for the Development of DAS (2016), p. 37.

[16] Christiansen, M.: In geheimer Mission: auf Abnahmefahrt mit der neuen Mercedes E-Klasse, W213 (2015).

[17] Winner, H. et al.: Freigabefalle des autonomen Fahrens (2010).

Table 1-1: Summary of the levels of driving automation.[18]

| Level | Name | Narrative definition | DDT | | DDT fallback | ODD |
|---|---|---|---|---|---|---|
| | | | Sustained lateral and longitudinal vehicle motion control | OEDR | | |
| **Driver performs part or all of the DDT** | | | | | | |
| 0 | No Driving Automation | The performance by the *driver* of the entire *DDT*, even when enhanced by *active safety systems*. | *Driver* | *Driver* | *Driver* | n/a |
| 1 | Driver Assistance | The *sustained* and *ODD*-specific execution by a *driving automation system* of either the *lateral* or the *longitudinal vehicle motion control* subtask of the DDT (but not both simultaneously) with the expectation that the *driver* performs the remainder of the *DDT*. | *Driver* and *System* | *Driver* | *Driver* | Limited |
| 2 | Partial Driving Automation | The *sustained* and *ODD*-specific execution by a *driving automation system* of both the *lateral* and *longitudinal vehicle motion control* subtasks of the *DDT* with the expectation that the *driver* completes the *OEDR* subtask and *supervises* the *driving automation system*. | **System** | *Driver* | *Driver* | Limited |
| **ADS ("System") performs the entire DDT (while engaged)** | | | *System* | **System** | *Fallback-ready user (becomes the driver during fallback)* | Limited |
| 3 | Conditional Driving Automation | The *sustained* and *ODD*-specific performance by an *ADS* of the entire DDT with the expectation that the *DDT fallback-ready user* is *receptive* to *ADS*-issued *requests to intervene*, as well as to *DDT performance-relevant system failures* in other *vehicle* systems, and will respond appropriately. | | | | |
| 4 | High Driving Automation | The *sustained* and *ODD*-specific performance by an *ADS* of the entire *DDT* and *DDT fallback* without any expectation that a *user* will respond to a *request to intervene*. | *System* | *System* | **System** | Limited |
| 5 | Full Driving Automation | The *sustained* and unconditional (i.e., not *ODD*-specific) performance by an *ADS* of the entire *DDT* and *DDT fallback* without any expectation that a *user* will respond to a *request to intervene*. | *System* | *System* | *System* | **Unlimited** |

Transferring the approach from approving a human driver for public traffic, i.e. the necessary theoretical and practical assessment during the driver's license test to an automated system is not feasible. For human drivers, the performed tests are limited to exemplary situations and it is assumed that a human who passed those tests and fulfills all other requirements to obtain a driving license (e.g. minimum age and therefore experience, received driving training, mental and physical suitability) can handle all other situations as well. However, there is currently no method existing to prove that those requirements and the implicated abilities based on experience and training are fulfilled by a technical system. Therefore, on the one hand, the test limitation to exemplary situations is not valid anymore. On the other hand,

[18] SAE: J3016: Taxonomy and Definitions for […] Vehicle Automated Driving Systems (2018), p. 19.

testing all possible situations is not feasible with existing methods (cp. subchapter 2.1).[19] Thus, new methods have to be developed for the validation and approval of HAV.

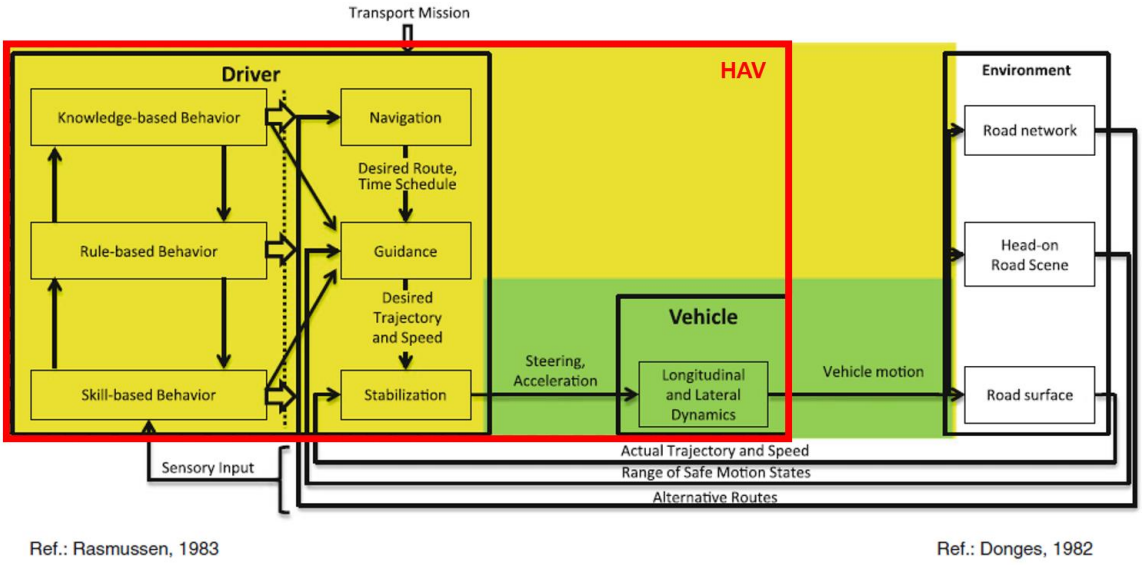


Figure 1-1: Illustration of the driving task taken over by HAV, based on the three-level model for human target-orientated behavior by Rasmussen[20] and the three-level hierarchy of the driving task by Donges[21]. [22]

## 1.2 Motivation and Research Theses

In order to develop new and standardized quality standards and methods for the validation of HAV, the research project PEGASUS (**p**roject for the **e**stablishment of **g**enerally **a**ccepted quality criteria, tools and methods as well as **s**cenarios and **s**ituations for the release of highly-automated driving functions) was launched 2016.[23]

Within PEGASUS, a scenario-based approach[24] is applied to reduce the validation effort for a highway chauffeur (an SAE level 3 system). It is assumed that the major part of mileage on the highway goes well without any special events while challenging scenarios are quite rare and randomly distributed in real traffic. Testing of the first-mentioned ordinary scenarios is without relevant contribution to the validation process. Therefore, the identification of

[19] Wachenfeld, W.; Winner, H.: The Release of Autonomous Vehicles (2016), pp. 437 f.

[20] Rasmussen, J.: Skills, rules, and knowledge […] in human performance models (1983).

[21] Donges, E.: Aspekte der aktiven Sicherheit bei der Führung von Personenkraftwagen (1982).

[22] Figure adapted from: Wachenfeld, W.; Winner, H.: The Release of Autonomous Vehicles (2016), p. 434.

[23] DLR: About PEGASUS - project homepage (2019).

[24] Scenario-based validation is handled in detail in section 2.1.1.2.

critical scenarios that can be reproduced in simulation or on test fields reduces the validation effort.

However, when testing the entire system as a black box that is only evaluated based on its observable behavior, some requirements on subsystem level are tested multiple times in different scenarios, as an evaluation on subsystem level is not intended. For instance, the correct reaction of an HAV to an obstacle in its lane has to be validated for all relevant types of obstacles when using a black box system test. Although for some subsystems (e.g. behavioral decision or trajectory planning and control) it does not matter, which kind of obstacle has to be handled as long as the obstacle (including state, position and dimensions) has been detected correctly. Thus, when decomposing the system into subsystems that are tested independently from each other and are evaluated on their interfaces, only the subsystems involved in detecting and classifying the obstacle have to be tested for different types of obstacles while the remaining subsystems are tested with an abstract obstacle. Additionally, not all conditions that influence the observable behavior of the HAV have an influence on each of its subsystems. Consequently, the following theses are set up and analyzed in this work:

**T 0:** *The validation effort for HAV can be reduced by functional decomposition and particulate testing.*

The necessary conditions for this are:

**T 1:** *A decomposition of highly automated driving functions into functional layers, which are independent of a concrete system architecture, is possible*.

**T 2:** *It is possible to define particulate test cases and related evaluation criteria that test each functional layer independently from the remaining system and evaluate it on its interfaces.*

## 1.3 Structure of the Dissertation

This dissertation is structured as follows:

The motivation and research theses are outlined in chapter 1, based on a short introduction of current test concepts and the so-called *approval trap*.

Thereafter, chapter 2 gives an overview of the state of the art and research in fields that are relevant for the topic. Firstly, existing approaches for the verification and validation of HAV are analyzed and compared with each other. Following that, an overview of the usage of the method of functional decomposition in general and its application to the driving task is given.

Based on the state of the art and the research theses set up in chapter 1, the objective of this dissertation is detailed in chapter 3. Therefore, the intended scope of the approach is defined. Thereafter the methodology to derive particulate test cases via functional decomposition is outlined. Furthermore, possible benefits of the approach as well as requirements for its development are derived. Finally, related research questions are identified.

The chapters 4 -7 deal with the methodology development for the application of functional decomposition as a validation method for HAV. Hereby, the research questions derived in chapter 3 are answered as follows:

- In chapter 4, functional layers and the related interfaces are defined based on the state of the art and the requirements on the validation method.
- Thereafter, chapter 5 deals with the derivation of particulate evaluation criteria for the individual tests of the functional layers.
- In parallel to that, in chapter 6, methods for the allocation of external and internal influence parameters to the functional layers are discussed.
- Thereafter in chapter 7, the generation of particulate test cases, based on the foundations laid out in the previous chapters, is outlined.

In chapter 8, the potential to reduce the approval effort by functional decomposition is quantified. Therefore, the parameter space explosion in scenario-based validation is analyzed and the required test coverage is estimated.

Eventually, a conclusion is drawn in chapter 9, therefore the results are compared with the requirements and expected benefits from chapter 3 as well as the motivation from chapter 1. Thereafter, the applicability of the approach in the automotive industry is discussed and remaining research questions are outlined.

This structure and the connections between the chapters are summarized in Figure 1-2.

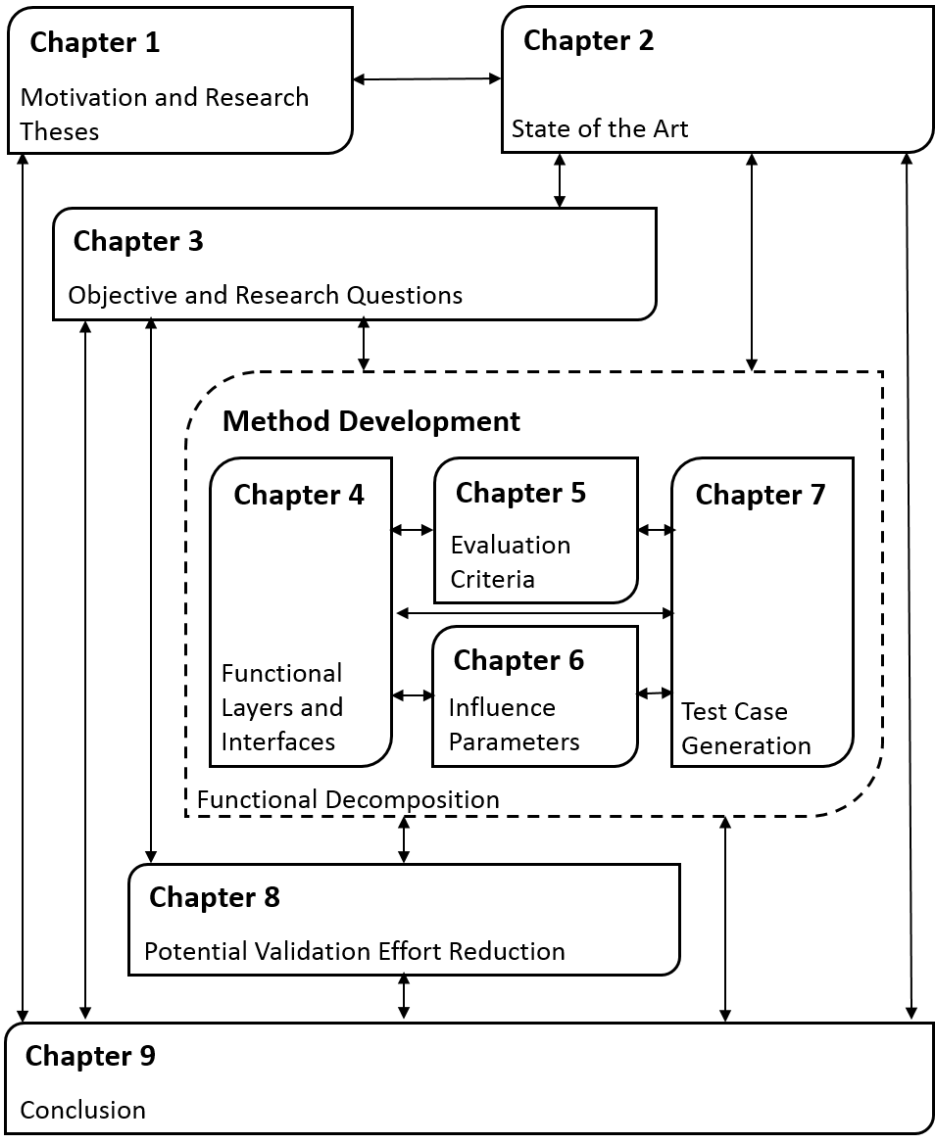


Figure 1-2: Structure of the dissertation.

# 2 State of the Art

In this chapter, an overview of the relevant state of the art in fields of interest for this dissertation is given.

## 2.1 Safety Verification and Validation of Highly Automated Driving

The following sections give an overview of the safety verification and validation of HAV. Hereby, it is differentiated between verification and validation approaches, test environments, evaluation criteria and test coverage.

In this work the following definitions for the central terms safety verification and safety validation are used:

- Safety verification is the "*determination whether or not an examined object meets its specified* [safety] *requirements*".[25a]
- Safety validation is the "*assurance, based on examination and tests, that the safety goals are adequate and have been achieved with a sufficient level of integrity*". [25b]

Thus, safety verification (in contrary to safety validation) does not include the proof whether the specified safety requirements are correct and complete. Therefore, safety verification is necessary, but not sufficient for the release of HAV.

### 2.1.1 Approaches for the Safety Verification and Validation of HAV

Junietz et al. [26] are giving an overview of the following existing approaches for the safety verification and validation of HAV and evaluate them:

- *Real-World Testing,*
- *Real-World Testing combined with Extreme Value Theory,*
- *Scenario-based Testing, and*
- *Formal Verification.*

[25] ISO: ISO 26262: Road vehicles – Functional safety (2018). Part I: Vocabulary. a: p. 24, b: p. 27.

[26] Junietz, P. et al.: Evaluation of Different Approaches to Address Safety Validation of AD (2018).

This classification is used to give an overview of existing approaches in the following section. As Junietz et al. [26] use the term *Real-World Testing* to describe an distance based approach that can be used in virtual environments as well, while other test approaches can also be conducted in the real world, the term *Distance-Based Testing* is used here instead. Furthermore, *Extreme Value Theory* (*EVT*) is considered as an approach for the reduction of validation effort, which is independent of *Real-World Testing* as it can be combined with other approaches as well. Therefore, *EVT* is handled in subchapter 2.1.5. Additionally, the class *Silent Testing* is included in this overview.

#### 2.1.1.1 Distance-Based Testing[27]

The idea behind distance-based testing is to estimate the average distance between two accidents - which can be used as a macroscopic safety reference[28] - if enough mileage is covered during testing. This statistical approach was first analyzed by Winner and Wolf[29]. This analysis was updated by Winner and Wachenfeld[30, 31] . Kalra and Paddock[32] analyzed the statistical approach for distance-based testing as well. To illustrate this challenge, the analysis of Wachenfeld and Winner[30] is summarized and updated to the most recent statistical data:

The objective of the following theoretical consideration is to prove that the object under test (OUT) is safer as a reference, e.g. today's traffic. Hereby the average distance between two accidents is used as a reference value. For this example, a HAV whose operational design domain (ODD)[33] is limited to the German Autobahn is used as OUT. Therefore, the statistical distance between two fatal accidents on the German Autobahn is used as reference distance $\bar{s}_{\mathrm{ref}}$ as it is the rarest (and therefore most difficult to prove) and most severe accident type. Additionally, the protection of human life is ethically important and enshrined in the constitutions of most countries, thus the number of fatal accidents is an essential reference in terms of legal and social acceptance. If less severe and more frequent accident types are used as a

[27] This section has been published in parts already in: Amersbach, C.; Winner, H.: Test Coverage for Scenario-Based Validation of HAV (2019).

[28] Cp. section 2.1.3.

[29] Winner, H.; Wolf, G.: Quo vadis, FAS? (2009), pp. 668–669.

[30] Winner, H.; Wachenfeld, W.: Absicherung automatischen Fahrens (2013).

[31] Wachenfeld, W.; Winner, H.: The Release of Autonomous Vehicles (2016), pp. 439–442.

[32] Kalra, N.; Paddock, S. M.: Driving to Safety: How Many Miles? (2016).

[33] According to SAE J3016 the ODD is defined as following: *"operating conditions under which a given driving automation system or feature thereof is specifically designed to function, including, but not limited to, environmental, geographical, and time-of-day restrictions, and/or the requisite presence or absence of certain traffic or roadway characteristics."*

reference, $\bar{s}_{\mathrm{ref}}$ becomes smaller. In 2017, 356 fatal accidents occurred on the Autobahn[34] while the total driven distance was 246 billion kilometers[35]. This leads to a reference distance of around 700 million kilometers. It is assumed that the occurrence of an accident is an independent and non-exhaustive random process and the distribution of accidents is a Poisson distribution. Furthermore, it is assumed that the OUT would be twice as safe as the reference and a significance level of 5 % is used. This leads to a distance factor of 10 that is required to reach a probability of success of 50 % for the test. Thus, under the above-mentioned assumptions, a test distance of around 7 billion kilometers would be required.[31]

Kalra and Paddock[32] are using statistical data from the US and different assumptions, however, their results are in the same order of magnitude. Both publications conclude that distance-based validation of HAV is not feasible before market introduction, due to the high required test distance.

#### 2.1.1.2 Scenario-Based Testing

As a distance-based validation of HAV is not feasible, amongst others Schuldt et al.[36] motivate a scenario-based validation approach, which is also the subject of various research projects [e.g. 37, 38, 39]. The main idea behind this approach is to generate relevant scenarios intentionally in simulations or on proving grounds rather than waiting until they randomly occur in public traffic.

##### 2.1.1.2.1 Terminology for Scenario-Based Testing

In this section, the most important terminology for scenario-based testing is summarized.

Ulbrich et al.[40] define a basic terminology for scenario-based testing:

> *"A **scene** describes a snapshot of the environment including the scenery and dynamic elements, as well as all actors' and observers' self-representations, and the relationships among those entities. Only a scene representation in a simulated world can be all-encompassing (objective scene, ground truth). In the real world it is incomplete, incorrect, uncertain, and from one or several observers' points of view (subjective scene)."*

[34] Destatis: Verkehrsunfälle - Zeitreihen 2017 (2018), p. 23.

[35] BASt: Fahrleistung von Kraftfahrzeugen auf Autobahnen 2017 (2019).

[36] Schuldt, F. et al.: Effiziente systematische Testgenerierung für FAS in virt. Umgebungen (2013).

[37] DLR: About PEGASUS - project homepage (2019).

[38] AVL LIST GMBH: About the project – Enable S3 (2019).

[39] Zhao, D. et al.: Accelerated Evaluation of Automated Vehicles Safety in Lane-Change Scenarios … (2017).

[40] Ulbrich, S. et al.: Defining […] the terms scene, situation, and scenario for automated driving (2015).

> *"A **situation** is the entirety of circumstances, which are to be considered for the selection of an appropriate behavior pattern at a particular point of time. It entails all relevant conditions, options and determinants for behavior. A situation is derived from the scene by an information selection and augmentation process based on transient (e.g. mission-specific) as well as permanent goals and values. Hence, a situation is always subjective by representing an element's point of view."*

> *"A **scenario** describes the temporal development between several scenes in a sequence of scenes. Every scenario starts with an initial scene. Actions&events as well as goals&values may be specified to characterize this temporal development in a scenario. Other than a scene, a scenario spans a certain amount of time."*

Ponn et al.[41] define the term *relevant scenario* as following:

> *"All scenarios that contribute to the type approval of automated vehicles are considered **relevant*** [for type approval]. ***Relevant scenarios** can also be very simple, such as the beginning of a speed limit. This is relevant for certification because an automated vehicle must comply with existing traffic regulations. […] A subset of the relevant scenarios are critical and complex scenarios."*

This definition can be transferred from type approval to validation or other applications.

***Complex scenarios***, according to Ponn et al.[41], are scenarios that are challenging for the planning algorithm, which is depending on the existence and movement of other traffic participants. According to Schuldt[42], the complexity of scenarios is defined by a multitude of dimensions, such as the number of elements, the number of states per element, novelty, openness of the target situation etc.

***Critical scenarios*** are scenarios that are close to accidents. They can be identified by criticality metrics such as TTC (time-to-collision). [41, 43]

The definition of relevant, complex and critical scenarios is illustrated in Figure 2-1.

---

[41] Ponn, T. et al.: Identify Relevant Scenarios for Type Approval of AV (2019), p. 3.

[42] Schuldt, F.: Diss., Ein Beitrag für den methodischen Test von autom. Fahrfunktionen (2017), p. 21.

[43] Wachenfeld, W. et al.: The worst-time-to-collision metric for situation identification (2016).

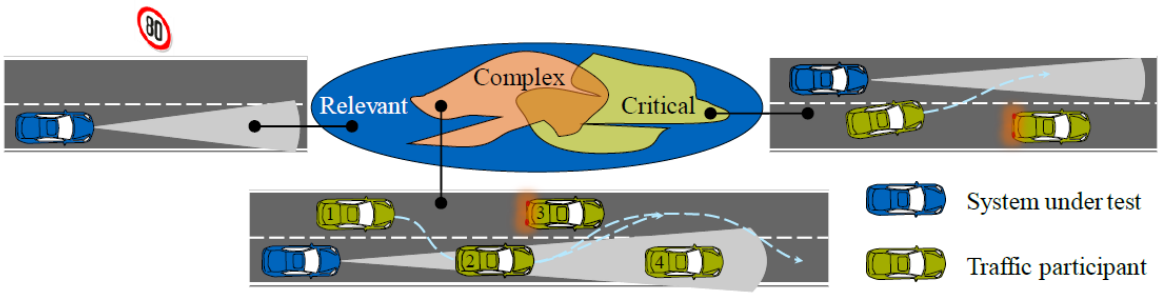


Figure 2-1: Definition of relevant, complex and critical scenarios.[41]

#### 2.1.1.2.2 Scenario description

Menzel et al.[44] define three levels of detail for the description of scenarios for different applications:

> *"**Functional scenarios** include operating scenarios on a semantic level. The entities of the domain and the relations of those entities are described via a linguistic scenario notation. The scenarios are consistent. The vocabulary used for the description of functional scenarios is specific for the use case and the domain and can feature different levels of detail."*
>
> *"**Logical scenarios** include operating scenarios on a state space level. Logical scenarios represent the entities and the relations of those entities with the help of parameter ranges in the state space. The parameter ranges can optionally be specified with probability distributions. Additionally, the relations of the parameter ranges can optionally be specified with the help of correlations or numeric conditions. A logical scenario includes a formal notation of the scenario."*
>
> *"**Concrete scenarios** distinctly depict operating scenarios on a state space level. Concrete scenarios represent entities and the relations of those entities with the help of concrete values for each parameter in the state space."*

Functional scenarios are the most abstract level of scenario description. They are used for item definition and hazard analysis in the concept phase according to ISO 26262[45]. Logical scenarios have a more detailed description and allocated parameter ranges within the parameter space. They are used to derive requirements during the system development phase. Concrete scenarios are allocated with concrete parameter values and are used to derive test cases during the test phase. Thus from functional over logical to concrete scenarios, the level of abstraction is decreasing while the number of scenarios is increasing.[46]

---

[44] Menzel, T. et al.: Scenarios for development, test and validation of automated vehicles (2018).

[45] ISO: ISO 26262: Road vehicles – Functional safety (2018).

[46] Menzel, T. et al.: Scenarios for development, test and validation of automated vehicles (2018).

Schuldt et al.[47] propose a 4-layered model for scenario description. This model is extended by Bagschik et al.[48] and Sauerbier et al.[49]. The model structures parameters that are required for the description of functional, logical and concrete scenarios into 6 layers, as can be seen in Figure 2-2.

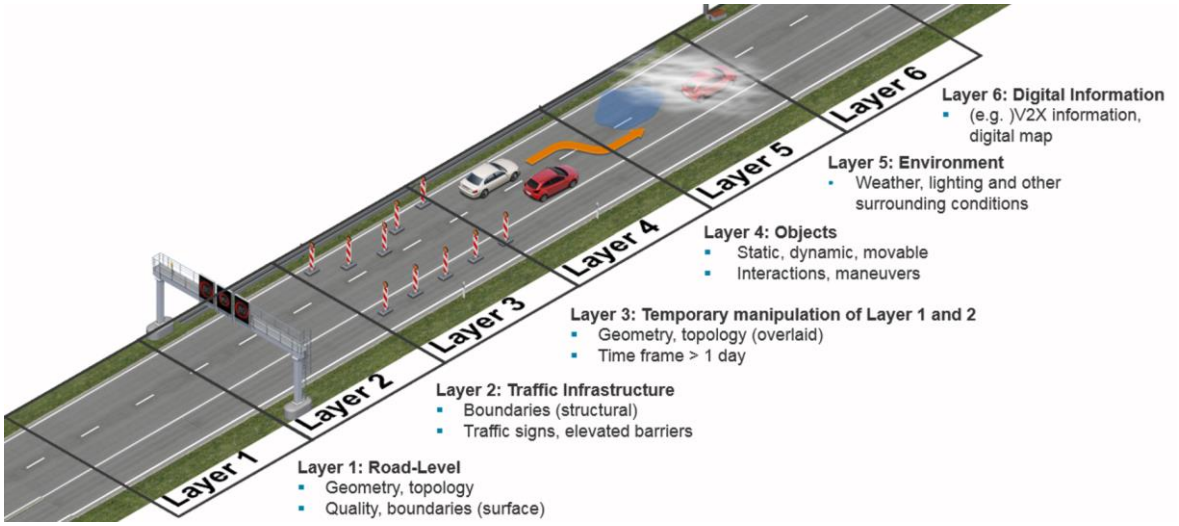


Figure 2-2: 6-layer model for scenario description.[50]

For the description of layers 1 to 3 in simulation environments, the standard data format OpenDRIVE® has been established. It is supplemented by the OpenCRG® format, which allows a more detailed description of road surfaces.[51] For the description of layers 4 and 5, the standard format OpenSCENARIO® was established.[52]

Menzel et al.[53] introduce a framework to transform a key-word-based description of functional scenarios into a description of logical scenarios in OpenDRIVE® and OpenSCENARIO®.

Although a common terminology and standardized data formats for scenario description exist, the range of the number of parameters that are used for scenario description as well as the used discretization step size is broad. Some examples for that fact can be found in Table 2-1.

[47] Schuldt, F. et al.: Effiziente systematische Testgenerierung für FAS in virt. Umgebungen (2013).

[48] Bagschik, G. et al.: Ontology Based Scene Creation for the Development of Automated Vehicles (2018).

[49] Sauerbier, J. et al.: Definition of Scenarios for Safety Validation of Automated Driving Functions (2019).

[50] Illustration by Schuldt, F. in: PEGASUS Project Office: PEGASUS METHOD (2019), p. 7.

[51] VIRES GmbH: OpenDRIVE® / OpenCRG® Product Data Sheet (2011).

[52] VIRES GmbH: OpenSCENARIO Homepage (2018).

[53] Menzel, T. et al.: From Functional to Logical Scenarios (2019).

Table 2-1: Variety of number of parameters and discretization steps of logical scenarios.

| **Reference** | **Type of scenario** | **Number of parameters *N*** | **Range of discretization steps per parameter** |
|---|---|---|---|
| Schuldt[54] | test scenario for a road construction assistant | 6 | 2 … 5 |
| ISO/PAS 21448[55] | example scenario for SOTIF safety analysis | 8 | 4 … 16 |
| Gao et al.[56] | test scenario for a lane departure warning | 16 | 2 … 8 |
| Amersbach, Winner[57] | exemplary Autobahn scenario "following" | 18 | 2 … 250 |
| Amersbach, Winner[57] | exemplary Autobahn scenario "swiss scenario" | 33 | 2 … 250 |

#### 2.1.1.2.3 Scenario Creation

Scenarios can be created either data-driven or knowledge-driven. With the data-driven approach, scenarios are extracted from recorded data from real traffic such as field operational tests, natural driving studies or accident databases by metrics or rule-based maneuver classification. [e.g. 58, 59, 60] Additionally, the data-driven approach can also be used with simulated traffic. Therefore, for example Hallerbach et al.[61] couple a traffic simulation and a vehicle dynamics simulation of the OUT and apply different criticality metrics to identify critical scenarios. Pütz et al.[62] propose a central database to store recorded data from real traffic clustered as logical scenarios including parameter distributions. Concrete scenarios are derived by sampling the parameter space. Waymo[63] is following a similar approach. They reproduce recorded scenarios in simulation while varying their parameters and thus create *"thousands of variations"* (i.e. concrete scenarios) of one single logical scenario.

[54] Schuldt, F.: Diss., Ein Beitrag für den methodischen Test von autom. Fahrfunktionen (2017), p. 122.

[55] ISO: ISO/PAS 21448:2019: SOTIF (2019). Annex F.

[56] Gao, F. et al.: A test scenario automatic generation strategy for intelligent driving systems (2019).

[57] Amersbach, C.; Winner, H.: Test Coverage for Scenario-Based Validation of HAV (2019).

[58] Junietz, P. et al.: Criticality Metric for the Safety Validation of Automated Driving (2018).

[59] Wachenfeld, W. et al.: The worst-time-to-collision metric for situation identification (2016).

[60] Krajewski, R. et al.: The highD Dataset (2018).

[61] Hallerbach, S. et al.: Simulation-based identification of critical scenarios (2018).

[62] Pütz, A. et al.: System validation of HAV with a database of relevant traffic scenarios (2017).

[63] WAYMO: On the Road to Fully Self-Driving (2017).

Menzel et al.[64] analyze different data-driven approaches and uncover their common weaknesses:

- Some parameters that are required for a complete and unambiguous scenario representation are not included in the recorded data, as they are not perceived by the measuring vehicle.
- For the variation of recorded scenarios, parameter dependencies[65], which are not included in the data, have to be considered and therefore manually added to the dataset.
- Only scenarios that have been recorded in the data set are considered. Thus, there is no evidence that all scenarios that can occur within the operational design domain are included.

Therefore, they propose to combine data-driven scenario creation with knowledge-driven scenario creation. To create scenarios with a knowledge-driven approach, relevant knowledge has to be collected and modeled, including semantic information. Bagschik et al.[66] propose to use ontologies for the representation of relevant knowledge from road traffic regulations, functional description of the OUT, guidelines, traffic sign and scenario catalogs as well as expert knowledge. These ontologies include dependencies between parameters and are used to create start scenes for validation scenarios by semantic combination. With this approach, they create over 40,000 scenarios for German motorways.[66] A comparison of both approaches can be seen in Figure 2-3.

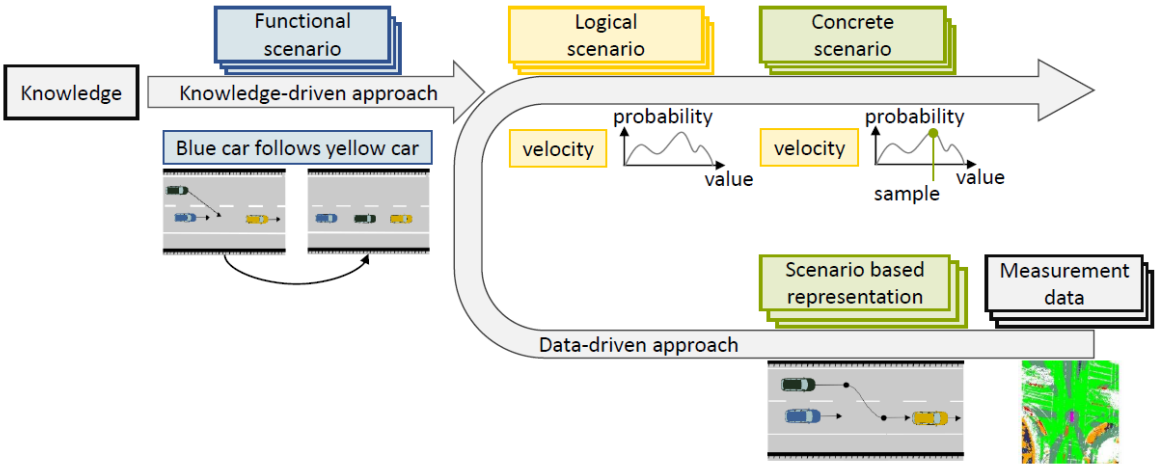


Figure 2-3: Comparison of data-driven and knowledge-driven scenario creation.[67]

[64] Menzel, T. et al.: From Functional to Logical Scenarios (2019).

[65] e.g.: when varying velocities in an overtaking scenario, the overtaking vehicle has to be faster than the overtaken vehicle.

[66] Bagschik, G. et al.: Ontology Based Scene Creation for the Development of Automated Vehicles (2018).

[67] Menzel, T. et al.: From Functional to Logical Scenarios (2019).

Büker et al.[68] are identifying automation risks, i.e. risks that are caused by automated vehicles. Therefore, they define three classes of automation risks:

1. Effects of the environment on the automated driving function.
2. Effects of the automated driving function on other traffic participants.
3. Driver interaction with the automated driving function and influence of the traffic environment.

The identification of automation risks is based on a HARA (hazard and risk assessment). The identified automation risks are used for knowledge-based scenario creation. The advantage of the approach is that automation risks are addressed, which is not the case for data-driven scenario creation that uses data from actual (non-automated) traffic.

Within the project PEGASUS, a holistic method for the scenario-based safety assessment of HAD functions has been developed. An overview of the 20-step method that uses both, data-driven and knowledge-driven scenario creation can be seen in Figure 2-4. [69, 70]

[68] Büker, M. et al.: Identifikation von Automationsrisiken hochautomatisierter Fahrfunktionen (2019).

[69] PEGASUS Project Office: PEGASUS METHOD (2019).

[70] PEGASUS Project Office: Description of the PEGASUS-Method (2019).

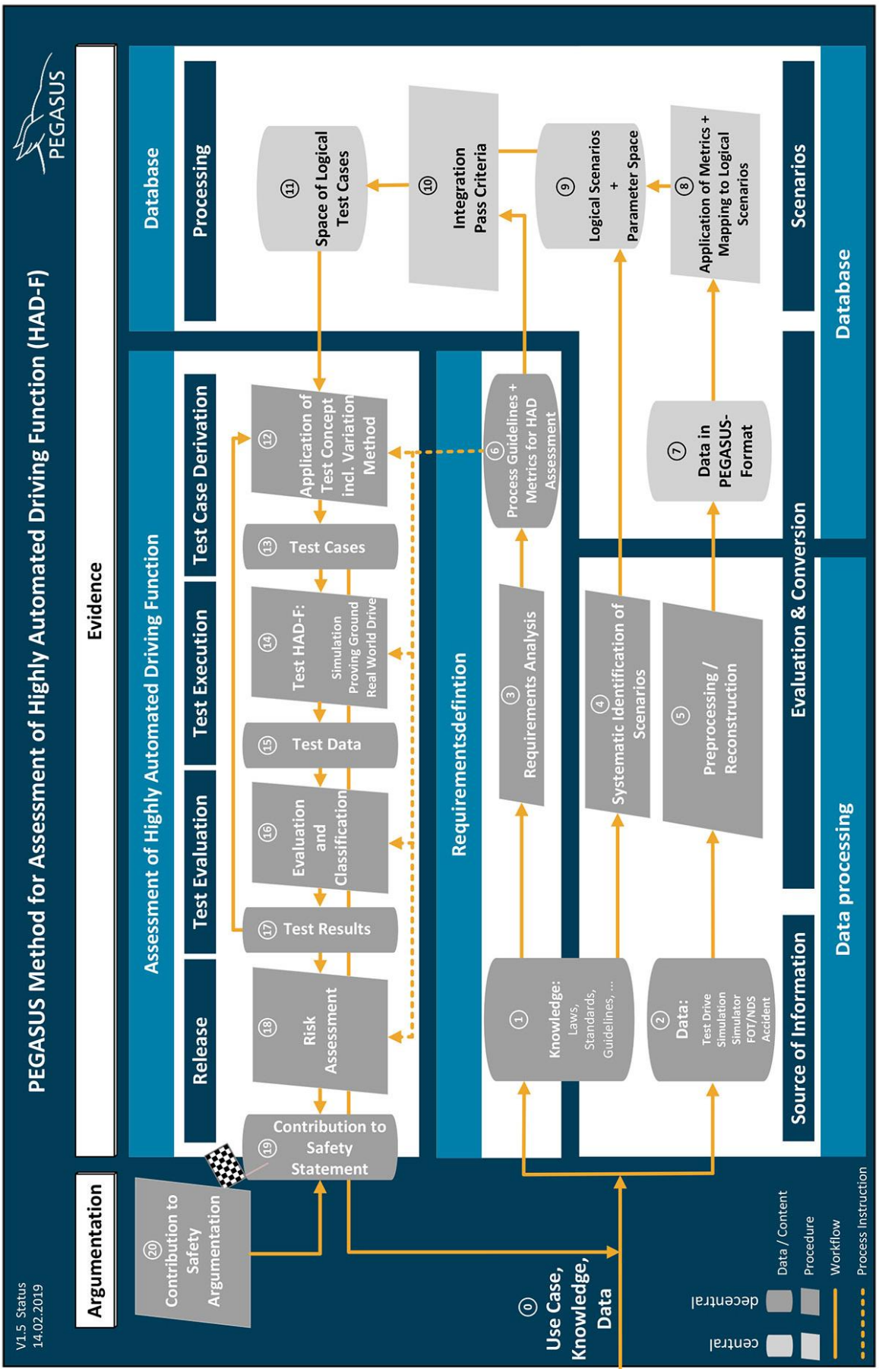


Figure 2-4: Overview of the PEGASUS method.[69]

#### 2.1.1.3 Silent Testing

Silent testing is an approach that tests the OUT, a HAD function, in an open-loop simulation using real sensor data in real traffic. Therefore, the test vehicle has to be equipped with the sensor set up of the OUT. The OUT is either evaluated online during the test drive, or the sensor data are recorded to carry out the open-loop simulation of the OUT after the test drive. During online silent testing, the OUT has no influence on the behavior of the test vehicle, which is controlled by a human driver or an already validated driving function. To evaluate the behavior of the OUT, it is compared to the behavior of the test vehicle.[71, 72] Alternatively, criticality metrics can be used for the evaluation.[73] There are different silent testing approaches documented in literature:

Winner[74] describes a silent testing approach for AEB (autonomous emergency braking) systems. Based on this approach, the VAAFO (virtual assessment of automation in field operation) method was introduced by Wachenfeld and Winner[71] and further developed by Junietz et al.[75]. The method uses differences between the trajectories of the OUT and the test vehicle as a trigger for event-based data recording. The recorded data are evaluated offline after the test drive. Hereby, the environment model of the OUT is corrected retrospectively to create a ground-truth environment model. As the simulation is open-loop, the behavior of other traffic participants cannot be influenced by the OUT and therefore only short time frames can be simulated when the behavior of the OUT is different to the test vehicle.[75]

König et al.[73, 76] propose a so-called “passive HAD” approach. They use offline trained driver models to allow reactive behavior of other traffic participants. Therefore, they use the recorded open-loop environment data as start scenes for a traffic simulation to close the loop. This and the following criticality assessment of the OUT with different criticality metrics are carried out online.

Tesla is applying a silent testing approach called “shadow mode”. They use customer vehicles in field operation rather than dedicated test vehicles to evaluate new driving functions before they become available via over-the air-update. Similar to the VAAFO approach, they compare the behavior of the OUT with the customer’s vehicle behavior. Found incidents are then fed back to the developers for further examination.[77]

---

[71] Wachenfeld, W.; Winner, H.: VAAFO a new runtime validation method (2015).

[72] Kalra, N.: Challenges and Approaches to Realizing AV Safety (2017), p. 5.

[73] Koenig, A. et al.: Passive HAD as a concept for validating HAV (2018).

[74] Winner, H.: Patent DE10102771: Einrichtung zum Bereitstellen von Signalen in einem Kraftfahrzeug (2000).

[75] Junietz, P. et al.: The Risk-Free VAAFO Tool (2019).

[76] Koenig, A. et al.: Bridging the gap between open loop tests and statistical validation for HAD (2017).

[77] Templeton, B.: Tesla's "Shadow" Testing (2019).

#### 2.1.1.4 Formal Verification

The aim of formal verification is to mathematically prove that the OUT is safe without performing any tests at all. Therefore, if a mathematical model of the system has been developed and validated, the approach would allow full coverage, contrary to other approaches.[78] Amongst others [e.g. 79, 80, 81], Shalev-Shwartz, S. et al from Mobileye[82] are working on formal verification by formalizing 5 common sense, high level accident avoidance rules to a *Responsibility-Sensitive Safety (RSS)* model. The requirement for the application of formal verification is a formalized regulatory environment. However, even if many traffic rules exist - which is the case at least in developed countries - the ruleset is to a certain amount fuzzy and not entirely formalized.[78] Furthermore, the approach assumes that all sensor input is correct and the software code has to be accessible. If all the requirements are met, it could be proven that the vehicle's behavior would be compliant with the rules. Assuming that all other traffic participants follow the rules as well, accidents should be prevented at all times. However, this covers only behavioral safety[83], other safety aspects e.g. component failures are not considered.[78, 84]

An approach for formal online (i.e. during vehicle operation in real time) verification of the trajectory planning of automated driving functions is proposed by Althoff and Dolan[85] and refined by Gruber and Althoff[86]. They use a reachability analysis approach to formally prove that all planned trajectories are collision-free.

### 2.1.2 Test Environments

Verification and validation tests for HAV can be carried out in virtual, partly virtual or real test environments. Partly virtual and virtual test environments are also known as X-in-the-Loop (XiL) methods. While tests that are carried out in environments with a high amount of virtual or artificial elements are most efficient financially and time-wise, the highest validity is reached with real test environments.[87] However, real systems are only subsets of reality

[78] Junietz, P. et al.: Evaluation of Different Approaches to Address Safety Validation of AD (2018).

[79] Kamali, M. et al.: Formal verification of autonomous vehicle platooning (2017).

[80] Mitsch, S. et al.: On provably safe obstacle avoidance for autonomous robotic ground vehicles (2013).

[81] Abbas, H. et al.: A Driver's License Test for Driverless Vehicles (2017).

[82] Shalev-Shwartz, S. et al.: On a Formal Model of Safe and Scalable Self-driving Cars (2017).

[83] See section 2.1.3.

[84] Ponn, T. et al.: Identify Relevant Scenarios for Type Approval of AV (2019).

[85] Althoff, M.; Dolan, J. M.: Online verification of automated road vehicles using reachability analysis (2014).

[86] Gruber, F.; Althoff, M.: Anytime Safety Verification of Autonomous Vehicles (2018).

[87] Wachenfeld, W.; Winner, H.: The Release of Autonomous Vehicles (2016), p. 444 ff.

and there are variations due to manufacturing tolerances amongst different exemplars of the system. Thus, testing only one single exemplar does not lead to high validity. The following paragraphs give an overview of the different classes and classification methods of test environments.

In virtual test environments, also known as SiL (software-in-the-loop), all elements that are required for the test execution are virtual i.e. simulated. Schuldt[88] gives a detailed overview of different abstraction levels of traffic simulations and existing simulation tools. Vehicle dynamics and traffic flow simulations have been under development for several decades and reached a level of accuracy that is valid for most applications, as long as the used models are correctly parametrized. However, the generally accepted validation of simulation models in general and environment perception sensor models in particular is still a challenge, as holistic validation methods and criteria are not available yet.[89] Besides the validation of models, the meaningful physical modeling of active environment sensors is a challenge itself.[90]

In XiL environments, some elements are real (except for SiL, where all elements are virtual), while others are virtual/simulated or artificial/emulated. An example for an artificial test environment is a proving ground, where situations are created artificially with real, artificial (i.e. “dummy targets”) or virtual object vehicles.[87] Stellet et al. [91] and Schuldt[88] are giving an overview of different XiL methods. The most common XiL methods are:

- Software-in-the-loop (SiL): equivalent to virtual test environments
- Hardware-in-the-loop (HiL): the OUT e.g. a sensor or a control unit is real, the other elements are artificial or virtual
- Driver-in-the-loop (DiL): real driver in a driving simulator
- Vehicle-Hardware-in-the-loop (VEHiL): real vehicle on a chassis dynamometer
- Vehicle-in-the-loop (ViL): real vehicle on a proving ground with an artificial or virtual environment and object vehicles.

In real test environments, all elements that are part of the test execution are real. Real test environments can be public roads or dedicated test beds[92] on public roads that are equipped with additional infrastructure such as vehicle-to-x-communication.

Wachenfeld and Winner[93] propose a classification of test environments based on the representation of vehicle and environment that can be seen in Figure 2-5.

---

[88] Schuldt, F.: Diss., Ein Beitrag für den methodischen Test von autom. Fahrfunktionen (2017). a: pp. 58–60; b: pp. 65 ff.

[89] Rosenberger, P. et al.: Towards a Validation Methodology for Sensor Models (2019).

[90] Holder, M. et al.: Challenges in Radar Sensor Modeling for Virtual Validation of AD (2018).

[91] Stellet, J. E. et al.: Testing of advanced driver assistance towards automated driving (2015).

[92] e.g. BMVI: Digital Test Beds (2019).

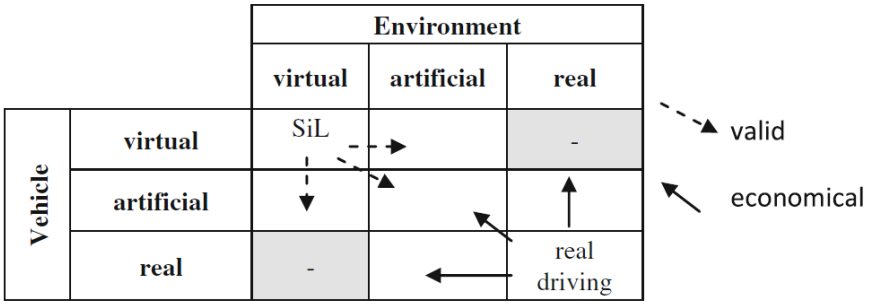


Figure 2-5: Classification of test environments according to Wachenfeld and Winner.[93]

Schuldt et al.[94] are further detailing the categories *vehicle* and *environment* to allow an unambiguous classification of all XiL environments. To illustrate this classification, they use Kiviat diagrams and assign the attributes real, simulated or emulated to the elements. This approach is extended by Steimle et al.[95] and can be seen in Figure 2-6.

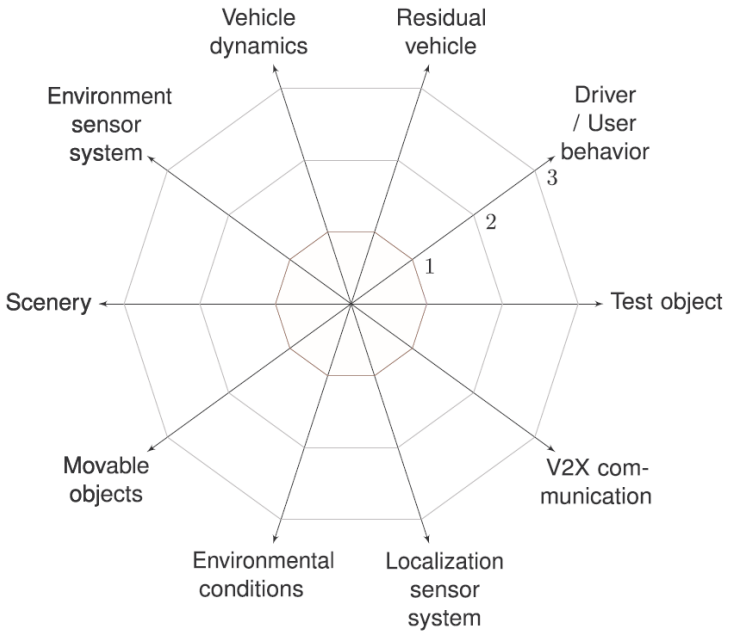


Figure 2-6: Classification of test environments according to Steimle et al.[95]; 1 = simulated, 2 = emulated and 3 = real.

[93] Wachenfeld, W.; Winner, H.: The Release of Autonomous Vehicles (2016), p. 444.

[94] Schuldt, F. et al.: Zuordnung von Testfällen auf X-in-the-Loop Verfahren (2015).

[95] Steimle, M. et al.: Classifying Test Bench Configurations (2019).

### 2.1.3 Evaluation Criteria and Safety Requirements[96]

Every validation approach requires evaluation criteria to evaluate whether a single test case has been passed or failed on a microscopic level and eventually whether the whole validation was successful on a macroscopic level. Those evaluation criteria in general are derived from requirements that have been defined during the product design process. Evaluation criteria for safety validation in particular are derived from safety requirements.

#### 2.1.3.1 Macroscopic Safety Requirements for Highly Automated Vehicles

Junietz et al.[97] are analyzing the risk acceptance of involved stakeholder groups in order to define macroscopic safety requirements (i.e. accident rate per mileage rather than specific requirements for individual driving situations) for HAV. For this purpose, they use the safety of today's traffic as a reference and apply concepts known from other fields, such as *ALARP* (as low as reasonably practicable), *MEM* (minimum endogenous mortality) and *GAMAB* (French: "*globalement au moins aussi bon*", generally at least as good as). They conclude that the risk acceptance of the different stakeholders depends on the market share of HAV. Starting at a market share of around 10 %, the safety requirement of society and non-users is assessed to be predominant over the safety requirement of the users and requires that HAV are at least 1.3 times as safe as today's traffic (on German Autobahn). With an increasing market share, the safety requirement level rises as well. Figure 2-7 shows the macroscopic safety requirements (i.e. the acceptable risk) for different stakeholders depending on the market share.

Liu et al.[98] are using a survey to determine risk acceptance rates for HAV. They conclude that HAV have to be 4 to 5 times as safe as today's traffic, using numbers from China without differentiating between different market shares.

Macroscopic safety requirements can be directly used as evaluation criteria for the distance-based approach.

---

[96] Parts of this section have been published already in: Klamann, B. et al.: Defining Pass-/Fail-Criteria (2019).

[97] Junietz, P. et al.: Macroscopic Safety Requirements for HAD (2019).

[98] Liu, P. et al.: How Safe Is Safe Enough for Self-Driving Vehicles? (2018).

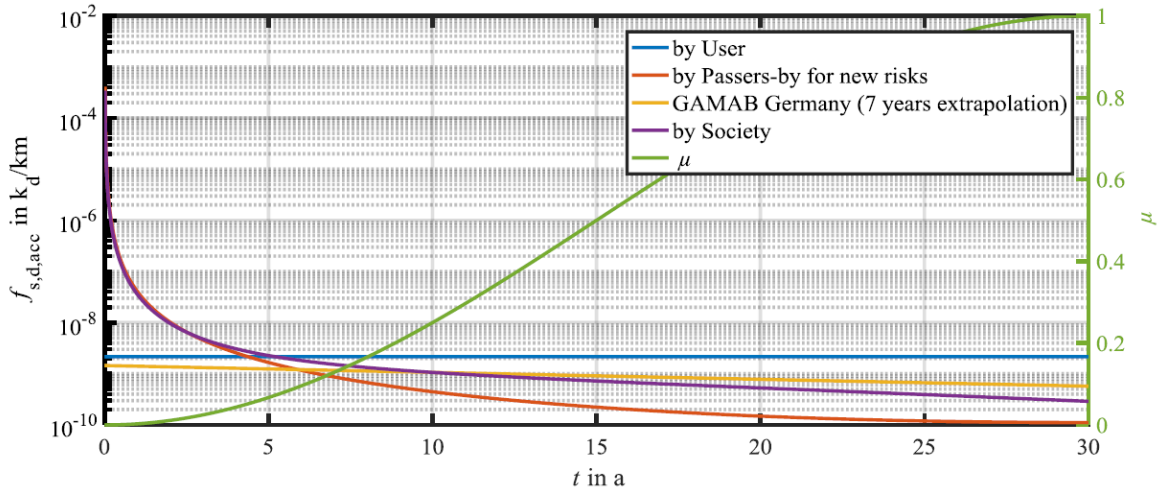


Figure 2-7: Macroscopic safety requirements for different stakeholders.[99] $f_{s,d,acc}$: acceptable risk; $k_d$: number of fatal accidents; μ: market share of HAV

### 2.1.3.2 Microscopic Safety Requirements

Microscopic safety requirements, i.e. specific safety requirements for individual vehicles, components or driving situations, can be defined for different safety aspects: [100]

***Behavioral safety*** focuses on how a system should normally behave in operation. In general, the behavior of a vehicle is expressed in its motion as well as interaction with traffic and its environment. Nolte et al.[101] define the term "external behavior" more specifically in order to describe a vehicle's visible behavior to influence surrounding traffic and therefore provoke hazards. Since "safety" implicitly means avoiding hazards, behavioral safety focuses on ensuring hazard-free visible actions of automated vehicles. Hence, the traffic rules provide a first regulatory framework for the behavior of automated vehicles, which is extended with further limiting factors by law. Within this regulatory legal framework, an automated vehicle must be capable of the driving tasks required by the present driving scenarios.

***Functional safety*** focuses on systematic and random failures of components and their handling. According to ISO 26262[102], a hazard analysis and risk assessment (HARA) has to be carried out to identify possible hazards and assign ASIL (automotive safety integrity level) based on their risk. To avoid "unreasonable" risk, safety goals (i.e. high-level safety requirements) are derived. The safety goals are decomposed into safety requirements for the affected components. For the HARA, different methods such as FTA (fault tree analysis) or FMEA (failure mode and effect analysis) are recommended by the ISO 26262[102] standard.

[99] Junietz, P. et al.: Macroscopic Safety Requirements for HAD (2019).

[100] WAYMO: On the Road to Fully Self-Driving (2017), p. 11.

[101] Nolte, M. et al.: Skill- and ability-based development process (2017).

[102] ISO: ISO 26262: Road vehicles – Functional safety (2018).

However, other methods have been applied successfully as well. Stolte et al.[103] for example use STPA (system theoretic process analysis) to derive safety goals and functional safety requirements for actuation systems for automated vehicles.

***Crash safety*** focuses on reducing the severity of collisions for the involved occupants and road users. Minimum crash safety requirements are defined by legislation authorities for type approval. Customer associations such as Euro /US NCAP define additional requirements that include test specifications and evaluation criteria. Those requirements are continuously extended to include new active safety measures of HAV.[104]

***Operational safety*** focuses on the interaction between vehicles and passengers. One example of an operational safety requirement is the avoidance of mode-confusion.[105]

***Non-collision safety*** focuses on hazards that could harm people that interact with the vehicle such as by an electrical shock during maintenance or after a crash.

In addition to the aforementioned safety aspects, ***cybersecurity*** has to be included in the validation of HAV, especially as security requirements can influence safety requirements and vice versa.[106] The definition of security requirements is done according to SAE J3061[107].

The remainder of this thesis focuses on behavioral and functional safety. It is assumed that the existing evaluation criteria for crash, operational and non-collision safety are still sufficient for automated vehicles, even though some details and concrete technical solutions (e.g. passenger restraint system for new interior concepts) need to be adapted.

### 2.1.4 Test Coverage[108]

The test coverage describes the “completeness” of a test suite. For distance-based testing, the test coverage can be specified as required distance under representative conditions (compare section 2.1.1.1). For formal verification, it is assumed that the complete environment is formalized which lead to full coverage without performing any test cases (compare section 2.1.1.4). For other test approaches, feasible coverage criteria have to be used.

---

[103] Stolte, T. et al.: Safety goals and functional safety requirements (2016).

[104] Gasser, T. M. et al.: Framework Conditions for the Development of DAS (2016), pp. 59–65.

[105] Winner, H.; Merkel, N. L.: Mode-Confusion und Inkompatibilitäten (2017).

[106] Schoitsch, E. et al.: The need for safety and cyber-security co-engineering and standardization (2016).

[107] SAE: Cybersecurity Guidebook for Cyber-Physical Vehicle Systems (2016).

[108] Parts of this section have been published already in: Amersbach, C.; Winner, H.: Functional Decomposition to Overcome the Parameter Space Explosion (2019).

Grindal et al.[109a] summarize test coverage criteria for combinatorial testing, i.e. testing of systems with multiple influence parameters that are applicable for scenario-based testing:

> *"**Each-used** (also known as 1-wise) coverage is the simplest coverage criterion. 100% each-used coverage requires that every interesting value of every parameter is included in at least one test case in the test suite.*
>
> *100% **pair-wise** (also known as 2-wise) coverage requires that every possible pair of interesting values of any two parameters is included in some test case. Note that the same test case may cover more than one unique pair of values.*
>
> *A natural extension of pair-wise (2-wise) coverage is **t-wise** coverage, which requires every possible combination of interesting values of t parameters be included in some test case in the test suite. t-wise coverage is formally defined by Williams and Probert[110].*
>
> *A special case of t-wise coverage is **N-wise** coverage, where N is the number of parameters of the test object. N-wise coverage requires that all possible combinations of all interesting values of the N parameters be included in the test suite."*

*N*-wise coverage is also known as exhaustive testing.

Although the *t*-wise coverage criterion is commonly used in combinatorial testing, the coverage level required for HAV is not defined yet. Gründl[111] states that traffic accidents are typically caused by a combination of several factors, following the so-called Swiss cheese model introduced by Reason[112]. Assuming failures of HAV are multicausal as well, 1-wise coverage is not sufficient. On the other hand, a scenario that is defined by $N$ parameters ($p_1,\ldots,p_i,\ldots,p_N$) with $k_i$ instances per parameter will lead to

$$S_N = \prod_{i=1}^{N} k_i \tag{2-1}$$

possible parameter combinations according to Grindal et al.[109b] Equation (2-1) often overestimates the number of required test cases for *N*-wise coverage as not all parameter combinations are existent in reality (e.g. snow-covered road in combination with high temperature). Nevertheless, *N*-wise coverage is not feasible for complex systems with a high number of influence parameters due to the progressive growth of the test suite.[113] Therefore Schuldt[114]

---

[109] Grindal, M. et al.: Combination testing strategies: a survey (2005). a: pp. 171–172; b: p. 169.

[110] Williams, A. W.; Probert, R. L.: A measure for component interaction test coverage (2001).

[111] Gründl, M.: Diss., Fehler und Fehlverhalten als Ursache von Verkehrsunfällen (2005), p. 19 ff.

[112] Reason, J.: The Contribution of Latent Human Failures to the Breakdown of Complex Systems (1990).

[113] Sommerville, I.: Software engineering (2006), p. 539.

[114] Schuldt, F.: Diss., Ein Beitrag für den methodischen Test von autom. Fahrfunktionen (2017), p. 124 f.

is proposing to use pair-wise or *t*-wise coverage (without recommending a concrete value for *t*).

Kuhn et al.[115] analyze empirical data from error reports in various domains and introduce the failure-triggering fault interaction (FTFI) number. The FTFI number is *"*[…] *the number of conditions required to trigger a failure"*. This means, that any failure with an FTFI number smaller than or equal to *t* will be discovered by testing with *t*-wise coverage. In the data analyzed by Kuhn et al.[115], the FTFI number does not exceed six. Figure 2-8 shows the cumulative FTFI number distribution for some of the analyzed domains.

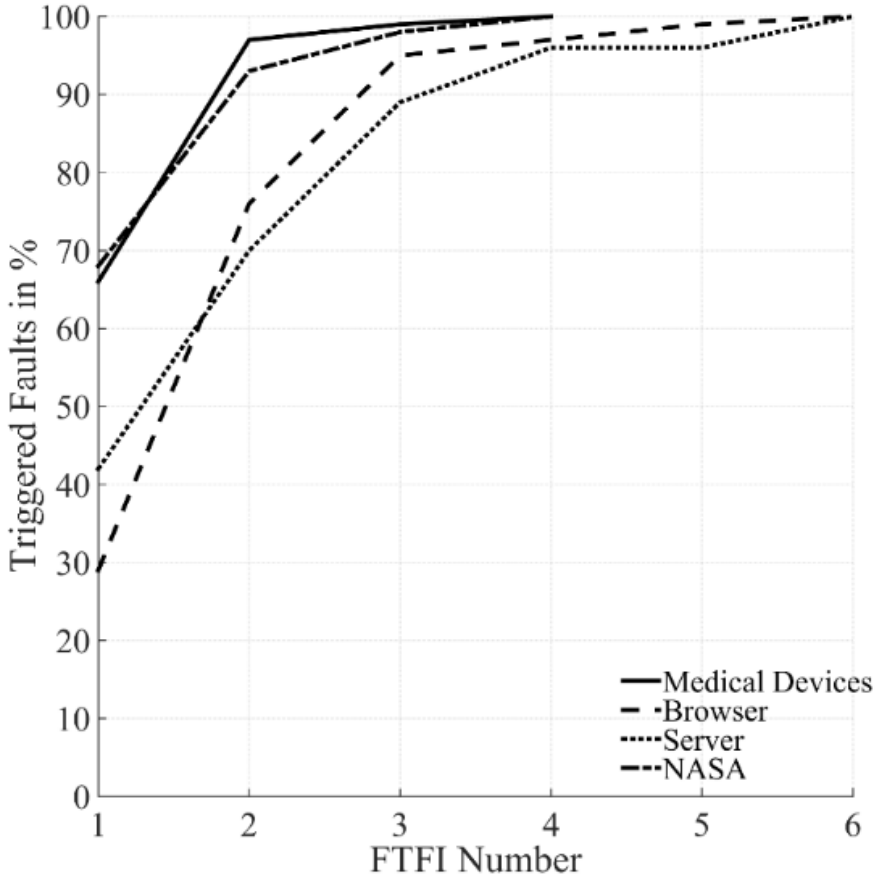


Figure 2-8: Fault rate for different FTFI numbers in different domains according to Kuhn et al.[115]

However, for HAV, empirical data do not exist yet, thus the FTFI numbers for such systems are unknown. Additionally, input parameters of scenarios are not necessarily comparable with faulty conditions in software systems.

Currently, there is no definition of the required test coverage for scenario-based testing, neither in the form of a number of test cases, nor in the form of a concrete coverage criterion. Therefore, the required test coverage for scenario-based validation of HAV is estimated in subchapter 8.2.

[115] Kuhn, D. R. et al.: Software fault interactions and implications for software testing (2004).

## 2.1.5 Approaches to Reduce Validation Effort

As one of the major challenges for the safety validation of HAV are the large test scopes required, this section gives an overview of different approaches that aim to reduce the approval effort.

### 2.1.5.1 Extreme Value Theory

Extreme Value Theory (EVT) is used in financial and insurance mathematics to extrapolate measurements to unlikely events that did not occur during the observation time. The approach can be used to estimate the occurrence probability of rare events, such as accidents, based on the probability distribution of critical scenarios (i.e. near-accidents). Thus, the required distance for distance-based validation can be reduced.[116] Åsljung et al. [117a] apply EVT on measured data from real traffic to extrapolate accident frequency from critical scenarios and compare their results with accident statistics. They conclude that if the correct criticality metrics and threshold values would be chosen, the required test distance could be reduced by a factor of 45.[117b] However, finding the right metric and threshold values is challenging without having existing ground truth data a-priori to fit the extrapolation to. Thus, it is not possible to validate metrics and threshold values for EVT for a new technology before introduction as no empirical data are available as ground truth yet. In theory, EVT can also be combined with scenario-based or silent testing to either reduce the number of test cases or increase the evidence created by a limited test coverage.

### 2.1.5.2 Scenario Selection and Reduction Methods

For scenario-based testing, an efficient selection of relevant scenarios is the key for reducing test effort, as any non-relevant scenarios in the test suite increase the test effort without creating valuable evidence for the validation process. However, the selection or creation of relevant scenarios is a huge challenge. According to Ponn et al. [118] also simple scenarios may be relevant while complex scenarios are not necessarily relevant and vice-versa. Additionally, the criticality of a specific logical or concrete scenario cannot be determined a-priori as the behavior of the OUT is unknown. Therefore, for example, a scenario that is expected to be uncritical from analyzing the initial scene can develop a high criticality or even an accident, if the OUT reacts other than expected for any reason.

---

[116] Junietz, P. et al.: Evaluation of Different Approaches to Address Safety Validation of AD (2018), p. 493.

[117] Åsljung, D. et al.: Using EVT for vehicle level safety validation and implications for AV (2017). a: -; b: p. 296.

[118] Ponn, T. et al.: Identify Relevant Scenarios for Type Approval of AV (2019), pp. 4–6.

Ponn et al.[118] analyze different scenario selection approaches and point out their strengths and weaknesses. This analysis is summarized in the following sections:[119]

**Design of Experiments** (Ahmed and Zamli[120]; Tatar[121] ; Schuldt et al. [122]; Grindal et al. [123])

Combination of parameters to meet certain coverage criteria e.g. *t*-wise coverage.
*+ Good parameter space coverage*
*- The selection of important parameters is difficult in advance*
*- No selection of test cases based on relevance*

**Analytic Hierarchy Process** (Saaty[124]; Xia et al.[125, 126])

Approach to detect important parameters including *expert-based analysis of key influence parameters.*
*+ Analytical method for the determination of relevant parameters*
*+ Creation of complex scenarios*
*- Requires expert knowledge*
*- Does not consider presumably simple scenarios that nevertheless lead to faulty behavior*

**Ontology-based knowledge-driven scenario selection** (Bagschik et al.[127]; Chen and Kloul[128]; Huelsen[129])

Expert knowledge regarding relevant elements and their relations are formally represented with ontologies. Those ontologies are used to automatically create scenes that are extended to scenarios.
*+ Elements defined in the knowledge base are also part of the test catalog*
*- Solely based on expert knowledge*
*- No evidence of the relevance of the defined scenarios for the proof of safety*

---

[119] Exact quotes from Ponn et al.[118] are marked in *italic font*.

[120] Ahmed, B. S.; Zamli, K. Z.: A review of covering arrays and their application to software testing (2011).

[121] Tatar, M.: Chasing Critical Situations in Large Parameter Spaces (2018).

[122] Schuldt, F. et al.: Efficient, systematic test case generation for ADAS in virtual environments (2018).

[123] Grindal, M. et al.: Combination testing strategies: a survey (2005).

[124] Saaty, T. L.: The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation (1980).

[125] Xia, Q. et al.: Automatic generation method of test scenario for adas based on complexity (2017).

[126] Xia, Q. et al.: Test scenario design for intelligent driving system ensuring coverage and effectiveness (2018).

[127] Bagschik, G. et al.: Ontology Based Scene Creation for the Development of Automated Vehicles (2018).

[128] Chen, W.; Kloul, L.: Ontology-based Approach to Generate ADAS Use Cases of Highway Traffic (2018).

[129] Huelsen, M.: Diss., Knowledge-based driver assistance systems (2014).

**Data-driven scenario selection** (Pütz et al.[130, 131])

Test scenarios are derived from recorded scenarios from real or simulated traffic with high criticality. The extracted scenarios are stored in a database.
*+ Inclusion of scenarios of various origins possible*
*- High storage requirements with nearly identical scenarios*
*- If the number of stored scenarios exceeds a manageable number, a method for selecting relevant scenarios is required again*

**Scenario selection based on accident databases** (Wang et al.[132])

Scenarios that lead to accidents are reconstructed in simulations. Parameters are Varied.
*+ Shows accident avoidance potential of the automated vehicle*
*- Detailed accident data required*
*- Provides only limited information on new risks and accidents introduced by the automated vehicle*

**Accelerated evaluation of automated vehicles** (Zhao et al.[133]; Huang et al.[134, 135, 136])

Frequency distributions of involved parameters are determined based on real driving data of certain maneuvers. The parameter distributions are adapted to create more "severe situations". Importance sampling theory is used to calculate the acceleration factor. Acceleration factors from 300 to 105 can be reached (i.e. each simulated kilometer corresponds to a real distance of 300 to 105 kilometers).
*+ Conversion from simulated to real-life traffic kilometers*
+ Comparability with macroscopic safety requirements (*e.g. kilometers per accident*)
*- Time-consuming and cost-intensive data collection*
*- Number of necessary involved parameters unknown*

---

[130] Pütz, A. et al.: Database approach for the sign-off process of highly automated vehicles (2017).

[131] Pütz, A. et al.: System validation of HAV with a database of relevant traffic scenarios (2017).

[132] Wang, L. et al.: Prospective safety assessment of HAD functions using stochastic traffic simulation (2017).

[133] Zhao, D. et al.: Accelerated evaluation of automated vehicles in car-following maneuvers (2017).

[134] Huang, Z. et al.: Evaluation of automated vehicles in the frontal cut-in scenario (2017).

[135] Huang, Z. et al.: Towards affordable on-track testing for autonomous vehicle (2017).

[136] Huang, Z. et al.: Sequential experimentation to efficiently test automated vehicles (2017).

**Adaption of existing logical scenarios with reachability analysis** (Althoff et al.[137, 138]; Manzinger et al.[139]; Gruber and Althoff[140])

> *Starting from a baseline scenario, the trajectories of road users are adapted in such a way that the planning of a safe trajectory for the automated vehicle becomes particularly challenging.*
> *+ Also suitable for online evaluation of the selected ego trajectory*
> - Tests are focused on trajectory planning

**Criticality optimization** (Tatar[141]; Koren et al.[142])

> *A concrete scenario is chosen as the starting point, executed in simulation and evaluated using a criticality metric. Subsequently, specific parameters of the scenario are varied and the change in criticality is evaluated* and optimized with classical optimization methods[141] as well as machine learning approaches[142] .
> *+ Criticality optimization during test execution ensures that critical scenarios are discovered*
> *- Numerous simulations of concrete scenarios required → particularly time and cost-intensive if high-fidelity simulation models for vehicle dynamics, sensors and environment are used*

Ponn et al.[143] finally propose an optimization method for an effective selection of relevant scenarios for the type approval of an automated vehicle. Hereby the system specification of the OUT is considered in order to select relevant scenarios that challenge the system-specific weaknesses.

As currently a lot of research for scenario-based validation of HAV in general and on scenario selection and reduction methods in particular is carried out, the overview by Ponn et al.[143] is not complete. Thus in the following sections, additional methods for scenario selection and reduction are summarized without claiming completeness.

Bach et al.[144] propose a two-step approach for the selection and reduction of scenarios that have been recorded in real traffic. In the first step, a classification tree is used to select relevant scenarios based on the OUT specifications. For example, geolocations or road categories are used in this step to preselect scenarios that are relevant for the ODD of the OUT. In

[137] Althoff, M.; Dolan, J. M.: Online verification of automated road vehicles using reachability analysis (2014).

[138] Althoff, M.; Lutz, S.: Automatic generation of safety-critical test scenarios (2018).

[139] Manzinger, S. et al.: Kooperative Bewegungsplanung autonomer Fahrzeuge (2017).

[140] Gruber, F.; Althoff, M.: Anytime Safety Verification of Autonomous Vehicles (2018).

[141] Tatar, M.: Chasing Critical Situations in Large Parameter Spaces (2018).

[142] Koren, M. et al.: Adaptive stress testing for autonomous vehicles (2018).

[143] Ponn, T. et al.: Identify Relevant Scenarios for Type Approval of AV (2019).

[144] Bach, J. et al.: Test scenario selection for system-level verification and validation (2017).

a second step, similar scenarios are filtered out by comparing the two-dimensional histograms of the significant inputs. Thus, the total number of scenarios is reduced without reducing the variety of the pair-wise input parameter combinations.

Langner et al.[145] refine the second step of this approach by replacing the filtering of redundant scenarios with manual comparison of histograms with an approach to estimate the uniqueness of recorded scenarios. They use *autoencoders*, neural networks that are commonly used for data compression and anomaly detection, to detect new scenes within a pool of recorded real-world test drives. As characteristic for the uniqueness of scenarios, a *novelty value* with respect to the existing scenario set is determined for each additional scenario. Scenarios with a low novelty value are not added to the set of relevant scenarios, as they are similar to already existing scenarios.

Abbas et al.[146] are proposing a game-in-the-loop approach to create critical scenarios for the test of camera-based perception and control layers of HAD functions. They use the video game *Grand Theft Auto V* as a world simulator that acts in real-time with the OUT, which is operated virtually in the game. Search and optimization algorithms are used to discover weather and operation conditions that lead to non-robust behavior.

Hallerbach et al.[147] introduce a generic simulation-based toolchain for the model-in-the-loop identification of critical scenarios in the early development phase. They use a vehicle dynamics simulation of the OUT that is coupled with a traffic simulation for other traffic participants. Two different metrics are used for criticality evaluation. Critical scenarios are flagged and the corresponding data is transferred back to the development department in order to improve the OUT.

### 2.1.6 Summary of the State of the Art for the Safety Verification and Validation of HAV

There are four main classes of verification and validation approaches for HAV described in the literature. Those approaches can be applied in different test environments as can be seen in Table 2-2.

While distance-based and scenario-based approaches can be applied in all test environments, silent testing can only be used in real test environments. However, XiL environments can be used to validate the silent testing approaches themselves. As formal verification does not require any tests, a combination with test environments is not applicable.

[145] Langner, J. et al.: Estimating the Uniqueness of Test Scenarios using Autoencoders (2018).

[146] Abbas, H. et al.: Safe at any speed: A simulation-based test harness for autonomous vehicles (2017).

[147] Hallerbach, S. et al.: Simulation-based identification of critical scenarios (2018).

Table 2-2: Overview of verification and validation approaches and test environments.

| + : compatible<br>- : not compatible<br>n/a : not applicable | | Test Environment | | |
|---|---|---|---|---|
| | | virtual | XiL | real |
| **Verification / Validation Approach** | distance-based | + | + | + |
| | scenario-based | + | + | + |
| | silent testing | - | - | + |
| | formal verification | n/a | | |

For all possible combinations of verification or validation approaches and test environments, a full test coverage for HAV does not look feasible due to the open-world problem with its high number of influencing parameters that lead to a parameter space explosion.

For a distance-based validation, which is state of the art for ADAS, macroscopic safety requirements have been derived from reference values from today's traffic and different risk acceptance theories. The required, but - at least in a real test environment - unfeasible, test coverage has been derived with a statistical approach.

Scenario-based validation is the focus of many research projects. Many cornerstones for the approach as well as a common terminology and standardized data formats are already available. However, there are still a lot of unsolved challenges, such as defining the required test coverage and parameter discretization as well as the selection of relevant scenarios. Nevertheless, scenario-based validation in virtual environments seems to be the most promising approach to overcome the validation challenge if valid simulation models for environment perception sensors become available.

Silent testing combines the advantages of testing in real environments (high validity) and testing in virtual environments (reduced costs and risk). If silent testing is used in existing vehicle fleets, e.g. customer vehicles or company fleets, there are no additional costs except for the initial costs for additional hardware and ongoing expenses for data transfer. Additionally, silent testing can be used for data-based scenario creation.

For formal verification, not all of the required preconditions are available yet and it is questionable if all made assumptions hold true. Nevertheless, if the method of formal verification can be applied eventually, no test cases would need to be conducted at all.

Except for distance-based testing in real environments, all methods seem to be feasible for an efficient validation of HAV if the remaining challenges are solved. However, as all remaining methods rely on some artificial or virtual elements and have limited test coverage, they have to be validated themselves. Therefore, Winner et al.[148] propose to conduct real-

[148] Winner, H. et al.: Validation and introduction of automated driving (2018), 190-192.

world driving tests to validate scenario catalogs. The frequency of surprises - i.e. unwanted conditions that are not included in the test catalog - per distance can be used as a measure for test catalog maturity. The inclusion of found surprises in the test catalog will then lead to an improvement of the catalog and the surprise rate will decline as illustrated in Figure 2-9. By extrapolating the trend line of the surprises per distance, the remaining risk can be estimated.

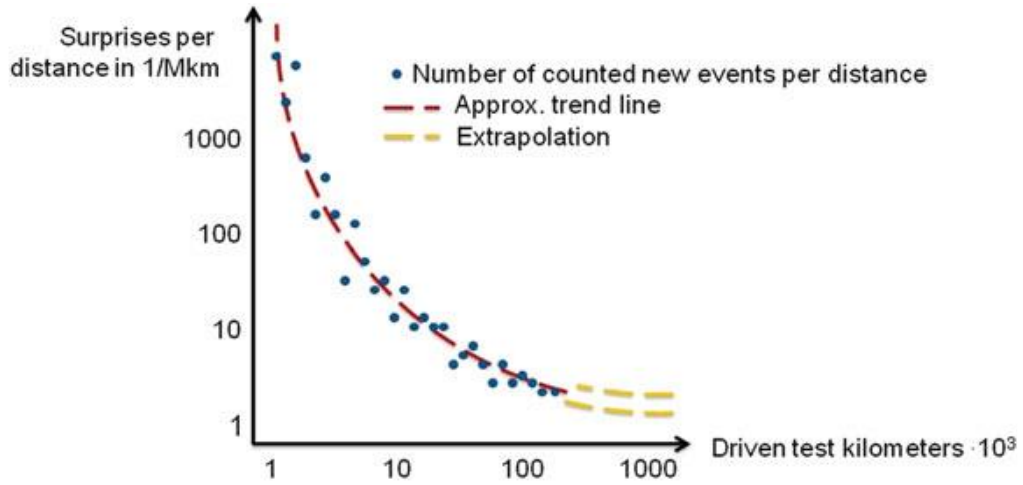


Figure 2-9: Number of surprises per distance covered.[149]

# 2.2 Functional Decomposition

As the objective of this dissertation is to develop a functional decomposition approach for the validation of highly automated driving functions, this subchapter gives an overview of decomposition in general and its application in automotive engineering as well as functional decomposition of the driving task in particular.

## 2.2.1 Decomposition in General

Decomposition is a common approach to reduce the complexity of systems or problems by dividing them into subsystems or subproblems respectively. The approach has successfully been used in computer science for several decades to increase computation efficiency.[150] Other common applications for decomposition are the approximation of partial differential equations[151] and optimization problems[152]. In addition to computer science and mathematics,

[149] Winner, H. et al.: Validation and introduction of automated driving (2018), p. 191.

[150] e.g. : Dantzig, G. B.; Wolfe, P.: Decomposition principle for linear programs (1960).

[151] Toselli, A.; Widlund, O.: Domain decomposition methods-algorithms and theory (2006).

[152] Cohen, G.: Optimization by decomposition and coordination: A unified approach (1978).

decomposition is also used in behavioral research [e.g. 153, 154] as well as in systems engineering to derive system architectures.[155] Analogous to the different viewpoints on system architectures as illustrated in Figure 2-10, there are also different forms of system decomposition:

- Functional Decomposition
- Capability Decomposition
- Hardware Decomposition
- Software Decomposition
- …

In product development, decomposition is used in the left branch of the V-model according to VDI 2206[156] to derive system architectures and requirements on subsystem and component level. Decomposition is a common method in automotive engineering as well. For example, the Automotive SPICE (Software Process Improvement and Capability Determination) standard recommends the use of functional and software decomposition for architectural design, requirements analysis and traceability. [157] In ISO 26262[158] decomposition is used to derive ASIL specifications for components starting from system or subsystem ASIL specifications.

[153] Reason, J.: Human error (1990).

[154] Rasmussen, J.: Skills, rules, and knowledge […] in human performance models (1983).

[155] Lotz, F. G.: Diss., Referenzarchitektur für die Fahrzeugführung (2017), pp. 45 ff.

[156] VDI: 2206: Entwicklungsmethodik für mechatronische Systeme (2004).

[157] VDA QMC: Automotive SPICE (2017).

[158] ISO: ISO 26262: Road vehicles – Functional safety (2018), Part 9, Section 5.

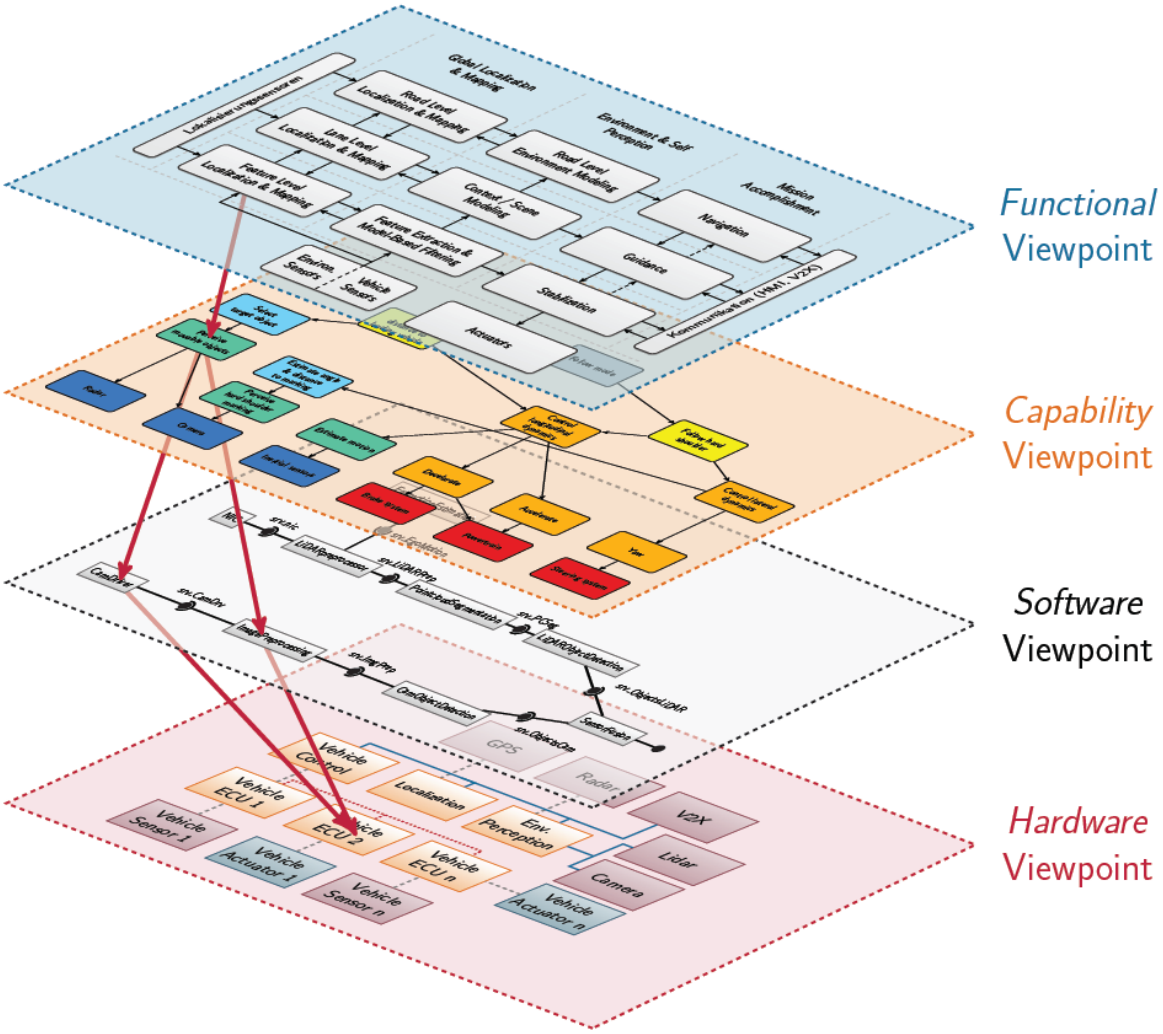


Figure 2-10: Different viewpoints on system architectures.[159]

## 2.2.2 Functional Decomposition of the Driving Task

In this section, the state of the art for the decomposition of the human driving task as well as the decomposition of automated driving functions is summarized.

### 2.2.2.1 Decomposition of the Human Driving Task

Decomposition of the human driving task for behavioral research and accident analysis has a long tradition.

[159] Bagschik, G. et al.: […] Architecture Framework for Safe Automated Vehicles (2018).

Donges[160] is decomposing the driving task into the three levels *navigation*, *guidance* and *stabilization*. Rasmussen[161] is introducing a three-level model for target-oriented human activities that is generally applicable to any kind of human work. The model originates from engineering psychology and is decomposing human behavior into the levels *knowledge-based behavior*, *rule-based behavior* and *skill-based behavior*. Both three-level models are combined by Donges[162] to model human driving behavior holistically as can be seen in Figure 2-11.[163a]

Endsley[164] is decomposing the process of dynamic human decision making in order to analyze situation awareness. The model that is intended to be used in various domains including aircraft control, systems operating and driving is illustrated in Figure 2-12.

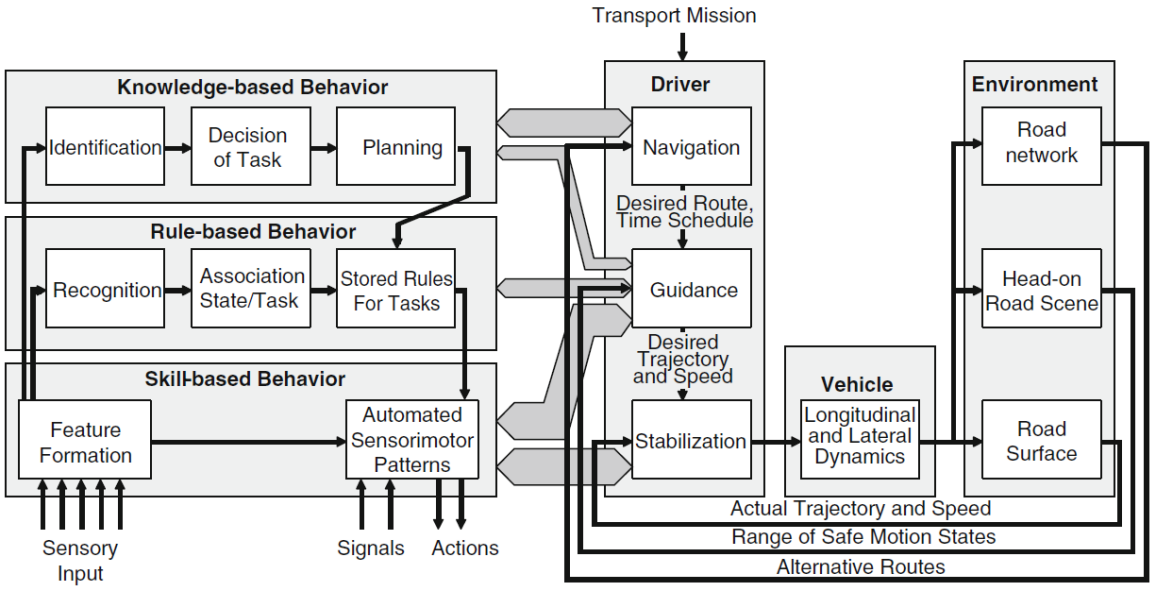


Figure 2-11: Categories of human target-oriented behavior[161] and three-level hierarchy of driving task[160] according to Donges[162].[163b]

[160] Donges, E.: Aspekte der aktiven Sicherheit bei der Führung von Personenkraftwagen (1982).

[161] Rasmussen, J.: Skills, rules, and knowledge […] in human performance models (1983).

[162] Donges, E.: Vorhersehbarkeit als Auslegungskonzept für Maßnahmen zur aktiven Sicherheit (1992).

[163] Donges, E.: Driver Behavior Models (2016). a: -; b: p. 21.

[164] Endsley, M. R.: Toward a theory of situation awareness in dynamic systems (1995).

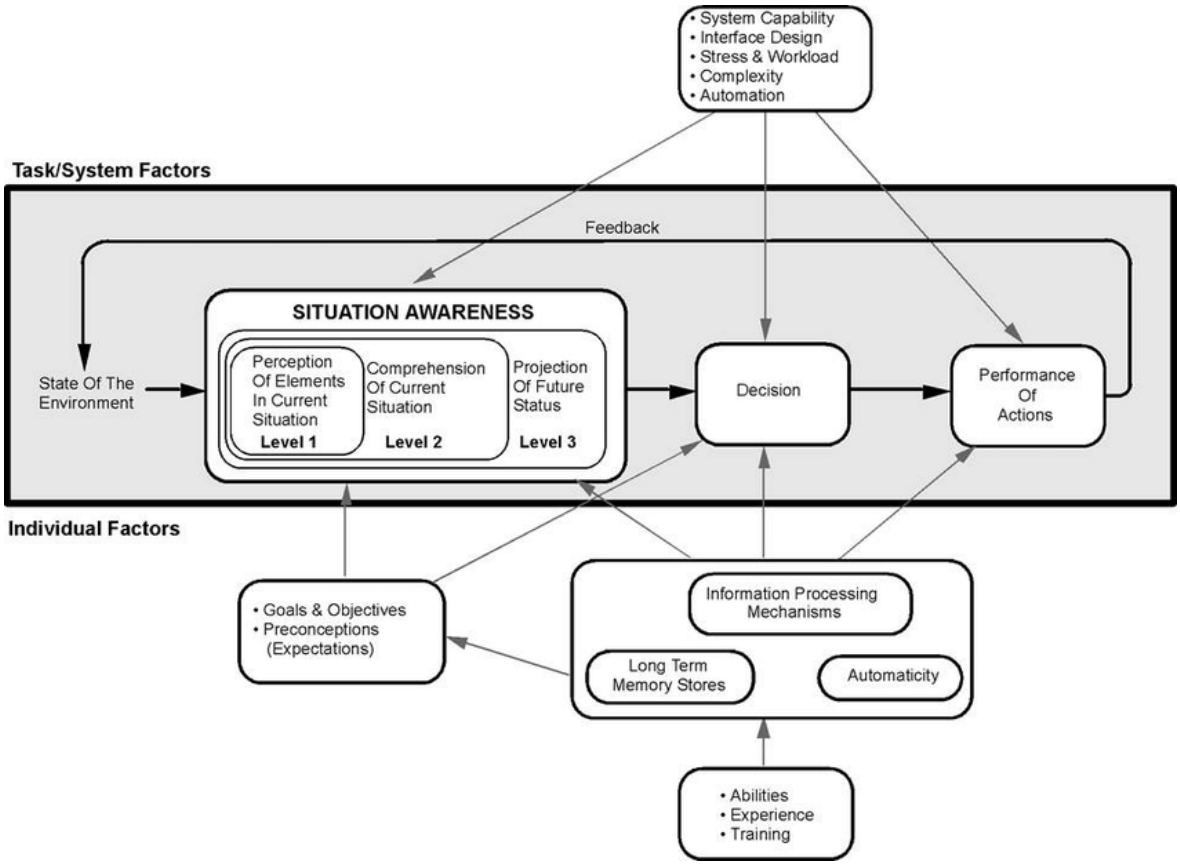


Figure 2-12: Endsley's model of situation awareness in dynamic decision-making.[165]

Gründl[166] is comparing action theoretic models for error classification in traffic accident analysis. All of them represent a decomposition of the human driving task. A summary of the models compared by Gründl can be found in Table 2-3.

[165] Illustration: Schmidt, G.: Diss., Influence of Warnings on Collision Avoidance Behavior (2012), p. 8.

[166] Gründl, M.: Diss., Fehler und Fehlverhalten als Ursache von Verkehrsunfällen (2005), pp. 80 ff.

Table 2-3: Summary of the comparison of action theoretic models for error classification by Gründl.[166]

| Name | Reference(s) | Description | Depic-tion |
|---|---|---|---|
| 4 layer model of information processing | Wickens[167] | Model to analyze mental operations in the action chain between stimuli reception and action execution. Attention and memory processes are included in the model as well as they influence reaction abilities as well as rule- or knowledge-based behavior. | Figure 2-13 |
| Model of internal malfunction | Rasmussen[168], Weigmann and Shappell[169], Zimmer[170] | Explicit algorithm to classify errors in the seqential action chain between information reception and action execution. Originally introduced by Rasmussen for all kind of human errors, the method was successfully adapted and applied to analyze aircraft accidents by Weigmann and Shappell. Zimmer adapted the schema for the analysis of traffic accidents. | Figure 2-14 |
| Model of unsafe actions | Reason[171] | The model separates unsafe actions in unintended and intended actions. Unindeted actions can have various reasons while intend actions are either based on wrong decisions or are violations (e.g. of speed limits). | Figure 2-15 |
| Error classification after Hacker | Hacker[172] | Combination and extension of the models by Rasmussen/Zimmer and Reason to a two-step error classification scheme. | Figure 2-16 |

[167] Wickens, C. D.: Engineering psychology and human performance (1992).

[168] Rasmussen, J.: Human errors (1982).

[169] Weigmann, D. A.; Shappell, S. A.: Human factors analysis of postaccident data (1997).

[170] Zimmer, A.: Wie intelligent darf/muss ein Auto sein? (2001).

[171] Reason, J.: The Contribution of Latent Human Failures to the Breakdown of Complex Systems (1990).

[172] Hacker, W.: Allgemeine Arbeitspsychologie (1998).

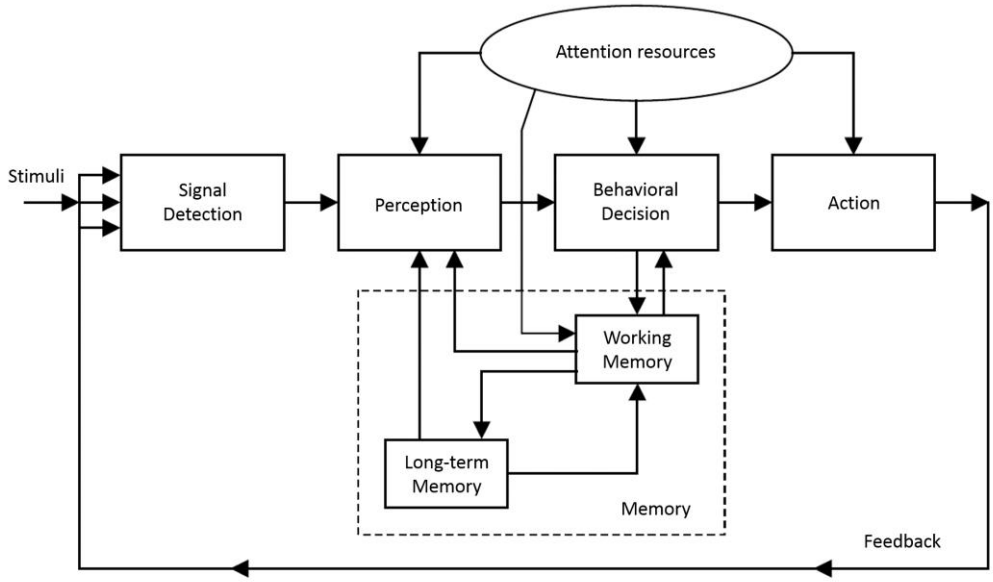


Figure 2-13: 4-layer model of information processing after Wickens. [173a]

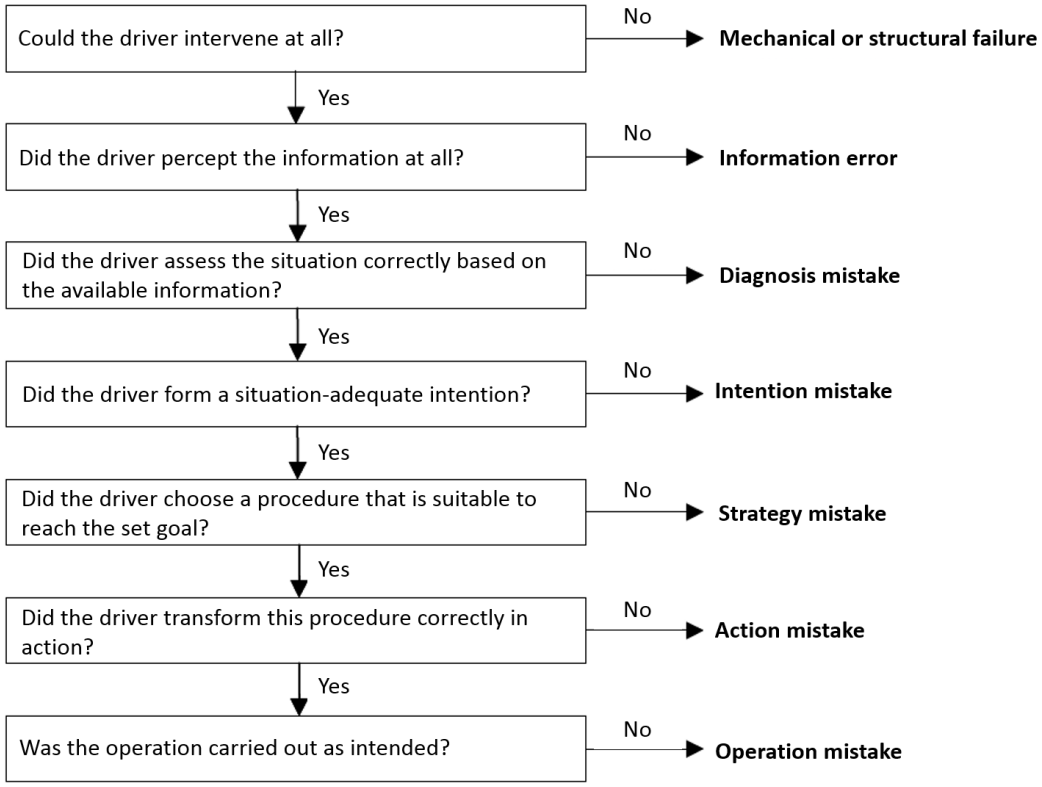


Figure 2-14: Model of internal malfunction by Rasmussen and Zimmer.[173b]

[173] Translated from Gründl, M.: Diss., Fehler und Fehlverhalten als Ursache von Verkehrsunfällen (2005).a: p. 80; b: p. 83.

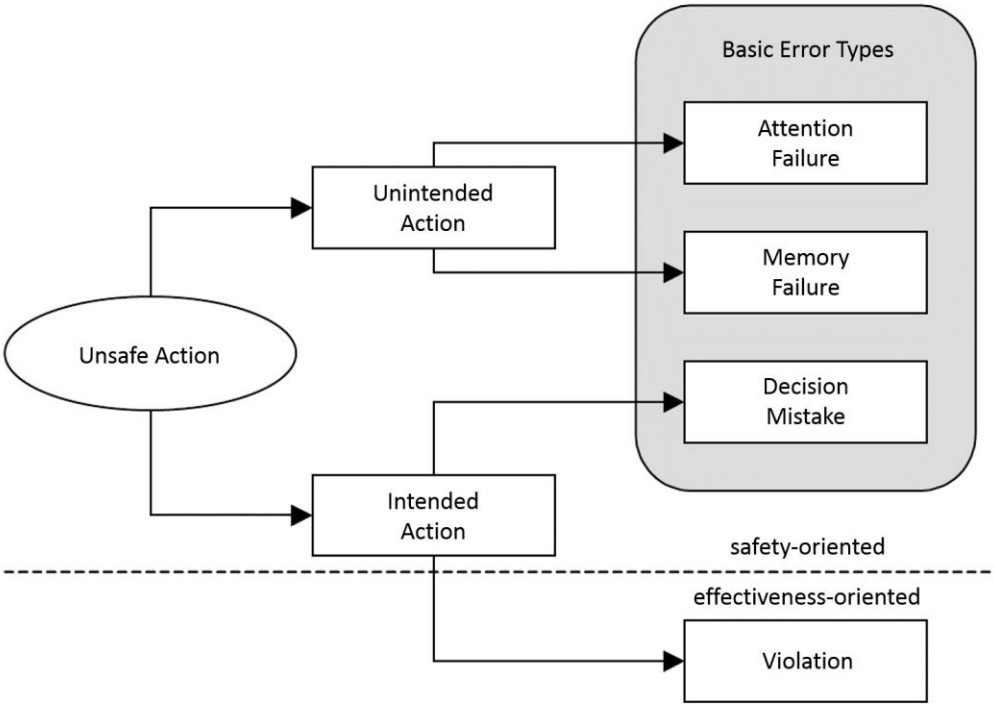


Figure 2-15: Model of unsafe actions after Reason. [174a]

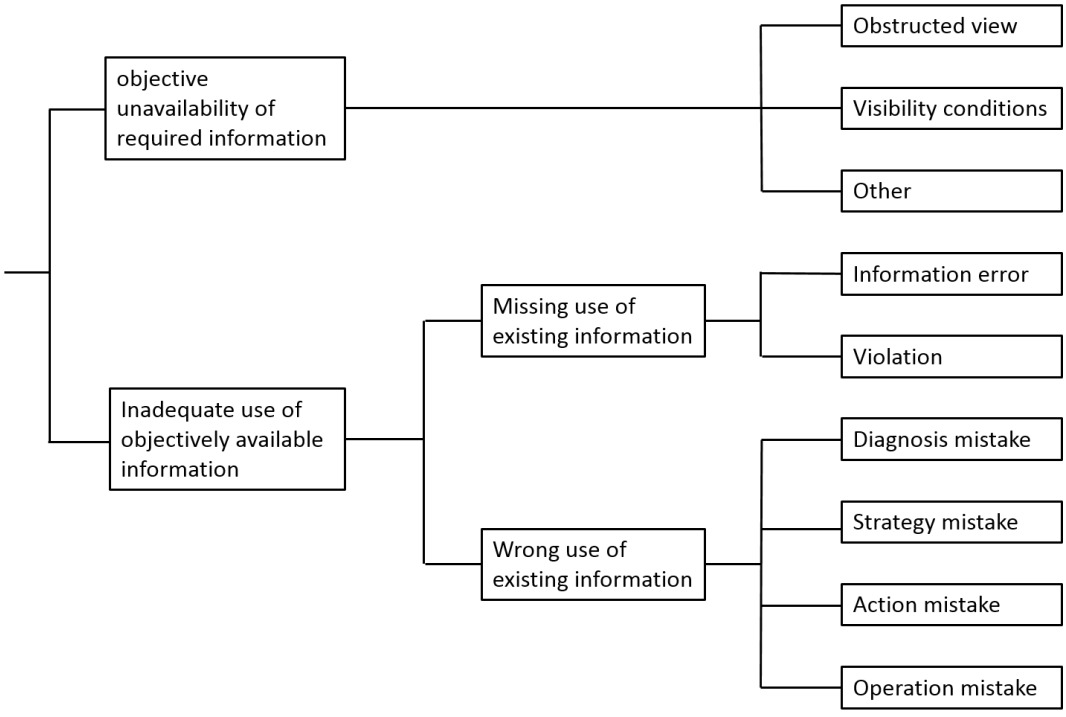


Figure 2-16: Error classification after Hacker.[174b]

[174] Translated and adapted from Gründl, M.: Diss., Fehler und Fehlverhalten als Ursache von Verkehrsunfällen (2005).a: p. 86 ; b: p. 91.

Graab et al.[175] are adapting the model of Rasmussen/Zimmer. They neglect the class of mechanical and structural failures as they focus on human errors. After analyzing 599 traffic accidents they conclude that strategy mistakes (i.e. using the brake pedal for an emergency brake instead of the theoretically also available handbrake) are not causing real accidents and therefore can be omitted. Additionally, they conclude that it does not bring any benefit to separate between action and operational mistakes, as the number of operational mistakes is too low. Furthermore, they separate information errors - which are the most common cause for accidents in the analyzed data set with a share of 65% - into impaired information access and impaired information reception. Thus, they use a 5-step decomposition of the human driving task with following decomposition layers:

(1) Information Access
(2) Information Reception
(3) Information Processing
(4) Behavioral Decision
(5) Action

Additionally, they add influence criteria and indicators for a more detailed error classification.

#### 2.2.2.2 Decomposition of Automated Driving Functions

Multiple functional system architectures for automated driving functions are documented in literature in different detail levels. Decomposition is commonly used to derive system architectures. Thus, system architectures can also be used as a decomposition scheme for driving functions. The *sense-plan-act* decomposition was one of the first system architectures to be applied in mobile robotics and first automated vehicles. It was dominant in the field until mid of the 80s when applications with higher complexity brought this simple architecture to its limits.[176] Nevertheless, *sense-plan-act* is still the most common functional high-level architecture for automated vehicles.[177]

Exemplary the more detailed functional system architectures by Lotz[178] (see Figure 2-17) and Matthaei[179] (see Figure 2-18) are referred here. Both can be mapped to the high-level *sense-plan-act* architecture that is further decomposed and extended with some kind of human-machine-interface (HMI). As the focus in Lotz's architecture is on cooperative automation, the HMI is further decomposed in its components. Whereas Matthaei's architecture

[175] Graab, B. et al.: Analyse von Verkehrsunfällen […] für die Entwicklung adaptiver FAS (2008).

[176] Gat, E. et al.: On three-layer architectures (1998).

[177] Anderson, J. M. et al.: Autonomous vehicle technology (2016).

[178] Lotz, F. G.: Diss., Referenzarchitektur für die Fahrzeugführung (2017).

[179] Matthaei, R.: Diss., Wahrnehmungsgestützte Lokalisierung (2015).

explicitly combines localization-based and perception-based approaches and thus the sense-layer that includes localization and external information such as digital maps, is put into focus. In contrast to simple hierarchical architectures that have - similar to decompositions of the human driving task (cp. 2.2.2.1) - a sequential information flow where the only feedback information flow is the loop-closure between the *act* and the *sense* layer via the vehicle's motion, the modules of both above-mentioned functional architectures are highly interconnected.

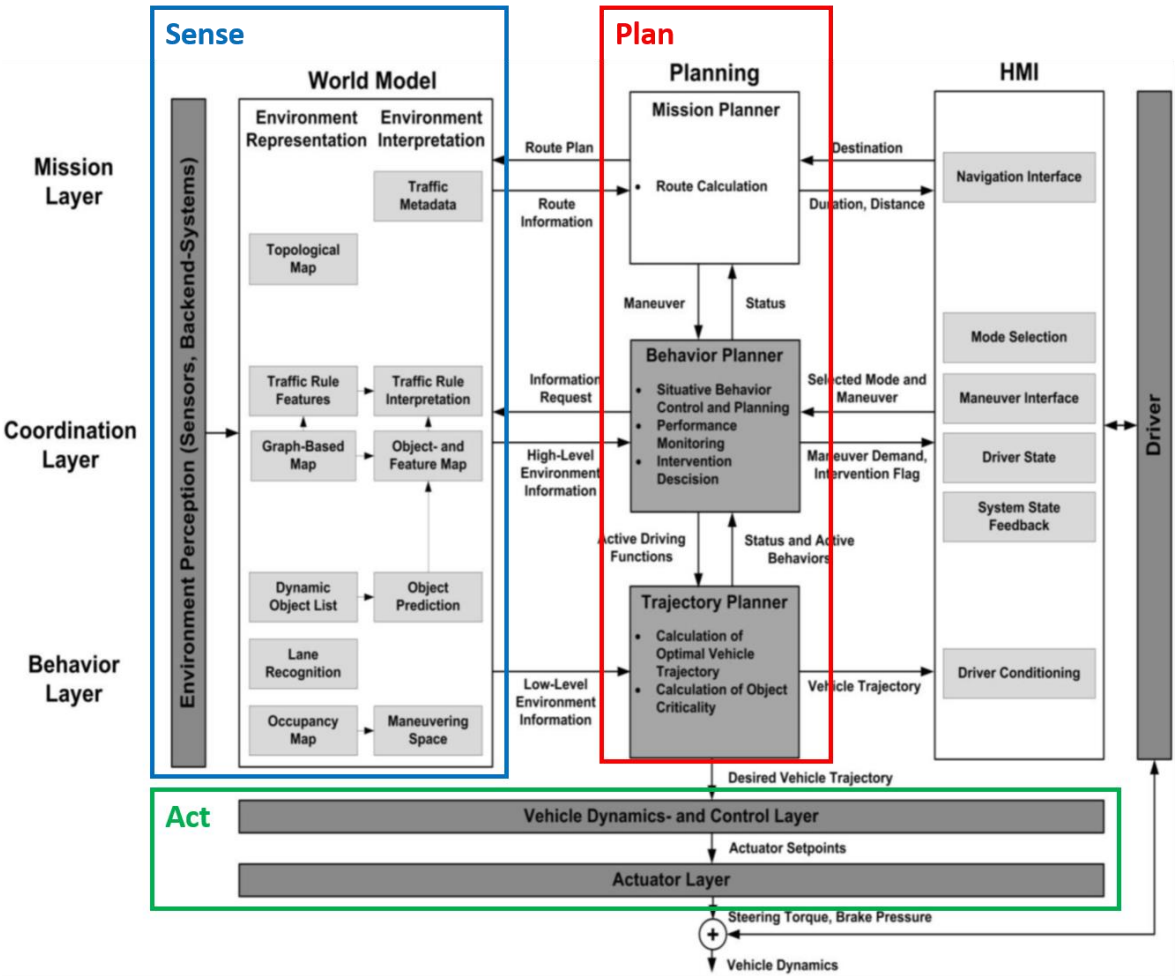


Figure 2-17: Reference architecture for scalable cooperative automation by Lotz.[180]

[180] Based on Hohm, A. et al.: Automated driving in real traffic (2014), p. 11.

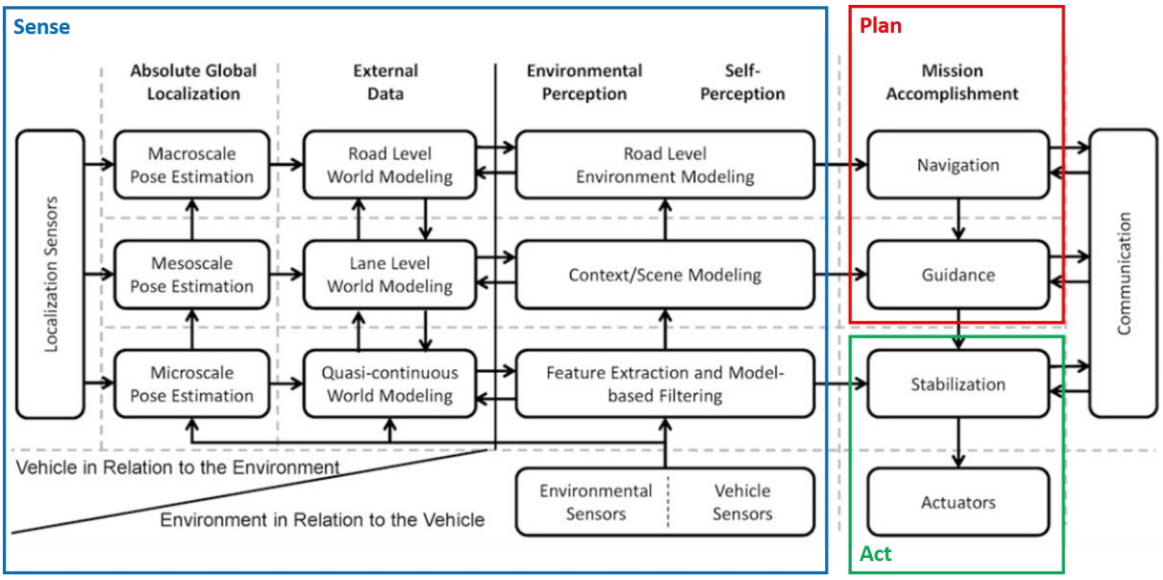


Figure 2-18: Functional system architecture for an autonomous on-road vehicle according to Matthaei and Maurer.[181]

Reschka[182] developed ability and skill graphs as a basis for the safe operation of automated vehicles. Compared to functional architectures, they describe the system from another viewpoint (cp. Figure 2-10), that is adapted from the actions of human drivers and represents another kind of system decomposition. Figure 2-19 shows a simplified ability graph of an adaptive cruise control (ACC).

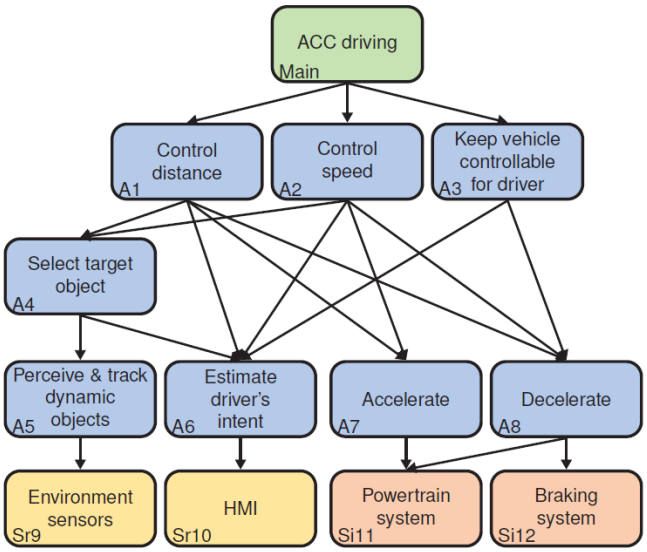


Figure 2-19: Simplified ACC ability graph.[183]

[181] Based on Matthaei, R.; Maurer, M.: Functional System Architecture (2018), p. 97.

[182] Reschka, A.: Diss., Fertigkeiten-und Fähigkeitengraphen (2017).

[183] Reschka, A. et al.: Ability and skill graphs (2015), p. 936.

The ability and skill graphs can be used for self-representation of the vehicle as well as for the derivation of safety goals and requirements. Furthermore, they can be used for the automated derivation of dependency chains (cp. subchapter 5.4).

### 2.2.3 Summary of the State of the Art on Decomposition

The concept of decomposition is not entirely new. It is used in various domains as for example in robotics, informatics, or mathematics to segment complex functions or problems into subfunctions or subproblems respectively. Other applications are behavioral research and systems engineering. Depending on the context, there are different viewpoints on decomposition, for example functional decomposition, hardware decomposition or software decomposition. Decomposition is broadly used also in automotive engineering for the derivation of system architectures as well as in requirements engineering.

Functional decomposition of the human driving task has a long tradition in behavioral research and accident analysis. There are various decomposition schemes described in literature. For automated or assisted driving, several decomposition schemes in the form of functional system architectures, all of which are extensions of the classical *sense-plan-act* architecture that originates from mobile robotics, exist.

# 3 Objective and Research Questions

In this chapter, the objective of this dissertation is concretized in a first step. Thereafter, the research questions that need to be answered to reach the set goal are derived.

## 3.1 Functional Decomposition as Safety Validation Method for HAD

The objective of this dissertation is to develop a method for the application of functional decomposition for the safety validation of HAV. Therefore, first of all the scope of the method is defined. Thereafter, the underlying methodology is specified, possible benefits are outlined and requirements are derived.

### 3.1.1 Scope

The scope of the method to be developed in this dissertation is the reduction of validation effort for HAV. Hereby, the focus is put on safety validation of the automated driving function itself. Thus, the method covers only behavioral and functional safety, as other safety categories are related to the base vehicle. Additionally, the safety requirements in other safety categories (e.g. crash safety) are not affected by the transition from assisted to automated driving and therefore existing safety validation methods can still be used (cp. section 2.1.3.2). Furthermore, system- and driver-initiated takeover situations are not covered, as there is already a lot of research that is focusing on takeover situations, which are only a very specific part of the operation scenarios.

In contrary to existing methods reducing approval effort (cp. section 2.1.5), it is not intended to optimize the selection of relevant scenarios but to increase the gained knowledge while omitting the test of irrelevant subspaces in the parameter space. Therefore, the common black-box system tests with evaluation of the observable behavior are replaced by grey-box system tests with evaluation at the standardized interfaces between the functional layers of the system. Hereby, the single functional layers can be tested individually if all possible input conditions - including system states - at the interfaces can be generated artificially by test harnesses[184].

[184] See section 6.3 for a detailed definition of the term “test harness”.

The creation and selection of scenarios as well as the definition of corresponding parameter spaces including the identification and discretization of influencing parameters are prerequisites for the method and thus not covered in this work. The same applies for the definition of macroscopic safety requirements (i.e. answering the question "*How safe is safe enough?*") as well as the definition of microscopic safety requirements on system level and specifying the required test coverage.

The method is not intended to derive safety requirements on component level by decomposing the systems requirements as it is proposed for example in ISO 26262[185] and Automotive SPICE[186]. Instead, the method aims to replace the traditional black-box system test by grey-box tests.

To summarize, the scope of this dissertation is to define a method to create *particulate*[187] test cases for the validation of functionally decomposed automated driving functions. This includes the definition of functional layers and interfaces as well as the derivation of particulate evaluation criteria and particulate parameter spaces as can be seen in Figure 3-1.

### 3.1.2 Methodology

An overview of the methodology to create particulate test cases is illustrated in Figure 3-1.

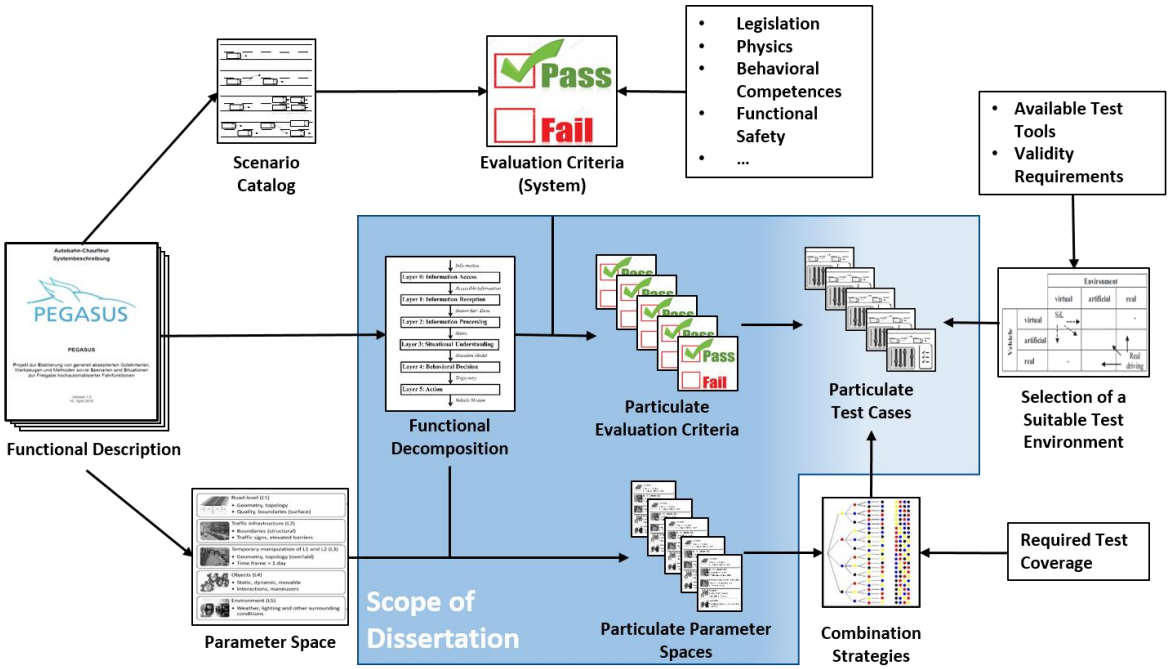


Figure 3-1: Methodology Overview.

[185] ISO: ISO 26262: Road vehicles – Functional safety (2018).

[186] VDA QMC: Automotive SPICE (2017).

[187] *Particulate* in this context means dedicated for one single functional layer.

The starting point is the functional description of the OUT. It includes either explicit or implicit – in the form of an ODD description - a catalog of relevant scenarios and the corresponding parameter space.

Based on the scenario catalog and requirements from various sources (e.g. legislation, vehicle dynamics, behavioral competences, functional safety requirements, etc.), evaluation criteria on system level are derived.

The automated driving function is decomposed into functional layers, based on its functional description. This functional decomposition of the OUT and the evaluation criteria on system level are used to derive particulate evaluation criteria for every single functional layer of the OUT. In parallel, the relevant parameter space is also decomposed in particulate parameter spaces.

Finally, a suitable combination strategy is used to create particulate test cases that fulfill the required coverage criterion. Moreover, a suitable test environment is chosen based on the available test tools and validity requirements.

The particulate test cases of the functional layers are part of the safety verification as they proof, that the particular evaluation criteria (i.e. safety specification) at their output interfaces are met. Nevertheless, the overall methodology finally leads to safety validation, if the complete parameter space of the particulate input parameters on each interface - as specified in the interface definition - are covert in the particulate tests. This is because the output of the final layer is than comparable to the observable output of the undecomposed system (see section 4.) that is traditionally evaluated during safety validation. Another prerequisite is, as for scenario based validation in general, that the scenario catalogue as well as the parameter space are compete and representative.

### 3.1.3 Intended Benefits[188]

Identifying critical scenarios can reduce the approval effort significantly. However, on some functional layers, the same abilities and requirements will be tested for several test cases. For instance, in the scenario “passing a static object”, it is assumed that the type of object is only relevant for the perception and information processing layers. For the decision and action layers, it does not matter if the object is for example a hedge or a guardrail. Furthermore, if a particular test fails, in contrary to a test of the complete system, the tests of the subsequent layers can be postponed until the failed test is finally passed.

If parts of existing HAD functions are used for new functions, the new method requires no or less re-testing in case of unchanged functional layers. This also applies to different variants of the OUT, which need individual approval with current test methods. To approve the

[188] This section is partly based on Amersbach, C.; Winner, H.: Functional Decomposition to Reduce Approval Effort for HAD (2017).

electronic stability control (ESC) of the Mercedes Sprinter for example, around 4500 different combinations of base vehicle, suspension, and load variants have been investigated according to Baake et al.[189]. In addition to the huge number of variants for the OUT, combinations of different parameters for a single scenario lead to high numbers of corresponding test cases. Lu[190] derives that around 43700 test cases for the scenario "lane change" would be required if the parameters "*initial speed*" and "*speed on the target lane*" were discretized in steps of 5 km/h within the parameter space and a "*pair-wise*" coverage as proposed by Schuldt et al.[191] is assumed. Variations of the environment (e.g. weather conditions), which are not considered by Lu, would lead to even higher numbers.

When testing the functional layers of the OUT separately as it is done with the new method, particulate tests from different scenarios can be aggregated if the test criteria/parameters are identical or if they are subsets of the criteria/parameter from another particulate test. Furthermore, the most suitable test tool (e.g. Simulation, XiL, test drive, etc.), depending on its validity, can be selected for each particulate test, which also helps to reduce the approval effort.

### 3.1.4 Requirements for Functional Decomposition as Validation Method[192]

To be able to develop a functional decomposition method for HAD functions, requirements on such a method have to be defined. Wachenfeld and Winner[193] state general requirements on a test case generation for the safety assessment of HAD functions:

> *"**Representative-valid**"*
>
> *"The requirement for representativeness has two aspects: On the one hand, the test case generation must ensure that the test coverage required is achieved. For example, a vehicle should not only be tested at 20 °C and sunshine, as it will be exposed to snow, rain and temperatures under 0 °C in real situations. Additionally, vehicle limit samples (tolerances during production) should be considered in the test case generation. On the other hand, the test execution must encompass the minimum degree of reality required. This means that the simplification in the representation of reality must not influence the behavior of the OUT nor the behavior and properties of the environment with respect to real behavior."*

[189] Baake, U. et al.: Versuchs-und simulationsbasierte Absicherung von ESP-Systemen für Transporter (2014).

[190] Lu, Y.: Masterthesis, Trajektorienplanung und Fehlerursachenanalyse in der Simulation (2015), p. 88.

[191] Schuldt, F. et al.: Effiziente systematische Testgenerierung für FAS in virt. Umgebungen (2013), p. 15.

[192] This section is partly based on Amersbach, C.; Winner, H.: Functional Decomposition to Reduce Approval Effort for HAD (2017).

[193] Wachenfeld, W.; Winner, H.: The Release of Autonomous Vehicles (2016), p. 433.

> *"**Economical**"*
>
> *"There are two parts to the requirement for the economical test concept: On the one hand, the test execution should be prepared and carried out as quickly as possible in order to be able to provide the persons involved in the development with feedback on the test object immediately. On the other hand, it must be ensured that the test execution is prepared and carried out at the lowest cost possible."*
>
> *"**Reproducible**"*
>
> *"Reproducibility greatly reduces the work required for regression tests. For example, if an error has been detected and the test object modified accordingly, the goal is to subject the OUT to a test in the same scenario as before."*
>
> *"**In good time**"*
>
> *"The earlier in the development process that a product can be tested informatively, the fewer the development steps that need to be repeated in the case of an error."*

In addition to the general requirements on a test case generation, there are requirements that are specific for the functional decomposition method, which will be defined below:

**Independent and generic decomposition layers**

In order to carry out the particular tests on different functional layers independently from each other, the functional layers have to be independent as well. Furthermore, the method should be applicable for different HAD functions and not be limited to functions that use a specific system architecture. Therefore, the defined decomposition layers have to be generic.

**Generic and observable interfaces between the functional layers**

Generic interfaces between the single functional layers are necessary to define the respective in- and output data for each layer. Those interfaces have to be observable to evaluate the particular tests.

**Explicit pass/fail criteria for all particular tests**

When carrying out a system test, it is straightforward to define pass/fail criteria. If for example an automated vehicle crashes into a static obstacle in a test case, this crash will be a fail criterion. However, if functional layers of the system are tested separately from each other, it is not that obvious anymore which criteria can be used to determine a pass or fail of the test. Therefore, explicit particulate pass/fail criteria are prerequisites for every test case.

## 3.2 Derivation of Research Questions

Having defined the objective and scope of this dissertation in the previous section, the research questions that are related to the theses set up in subchapter 1.2 are derived in this subchapter.

As a prerequisite for a functional decomposition of an automated driving function, functional layers and related interfaces have to be defined. As outlined in section 3.1.4, the decomposition layers as well as the interfaces have to be generic and independent. Furthermore, the interfaces have to be observable. This leads to the first research questions that are related to T 1: *A decomposition of highly automated driving functions into functional layers, which are independent of a concrete system architecture, is possible*.

**Q 1:** *Which independent and generic functional layers are suitable for a functional decomposition of HAD functions?*

**Q 2:** *How can the interfaces between the functional layers derived in Q1 be defined generically?*

The following research questions are related to

T 2: *It is possible to define particulate test cases and related evaluation criteria that test each functional layer independently from the remaining system and evaluate it on its interfaces.*

In order to create particulate test cases, particulate evaluation criteria have to be derived from system-level evaluation criteria. This leads to research question 3:

**Q 3:** *How can particulate evaluation criteria for decomposed HAD functions be derived?*

In order to derive particulate parameter spaces, the single influence parameters have to be assigned to the functional layers. This leads to the next research question:

**Q 4:** *How can influence parameters be allocated to the functional layers derived in Q1?*

After having created the prerequisites, particulate test cases can be derived. This leads to research question 5:

**Q 5:** *How can particulate test cases be defined?*

Finally, the top thesis

T 0: *The validation effort for HAV can be reduced by functional decomposition and particulate testing.*

has to be confirmed. Therefore, the potential to reduce the validation effort with functional decomposition has to be quantified. This leads to the final research question:

**Q 6:** *How high is the potential for reducing the validation effort of HAV by functional decomposition?*

The research questions derived here are going to be answered in the following chapters.

# 4 Functional Layers and Interfaces[194]

In this chapter, the first two research questions are addressed and the functional layers for the decomposition of HAD functions and the appropriate interfaces are defined.

## 4.1 Definition of Functional Layers

This subchapter addresses research question one:

Q 1:*Which independent and generic functional layers are suitable for a functional decomposition of HAD functions?*

In section 2.2.2, various existing decomposition schemes for human and automated driving tasks are summarized. In section 3.1.4, the requirements for functional decomposing as a validation method are derived. The specific requirements for defining the functional layers are independence and generality of the layers.

To fulfill the requirement of independence, a decomposition scheme without complex interdependencies between the single layers or modules has to be chosen. Therefore, the functional system architectures proposed by Lotz[195] and Matthaei[196] (cp. section 2.2.2.2) are not suitable as a decomposition scheme in this work. As they are on a too detailed level, they have a lot of interconnections and interfaces that would need to be generically modeled and standardized. Furthermore, they violate the requirement of generality as they include specific modules, i.e. "*graph-based map"* or "*lane-level world modeling"*, which are not necessarily components of any driving function. Another problem with such detailed decomposition schemes with many layers would be to derive explicit particulate evaluation criteria for each of those layers. However, both system architectures will be used as references to demonstrate the generality of the defined functional layers and interfaces (cp. subchapter 4.3.)

The ability and skill graphs by Reschka[197] are also too detailed and interconnected to serve as a basis for a functional decomposition scheme in this thesis. Nevertheless, the concept can help to derive particulate evaluation criteria (cp. chapter 5).

---

[194] This chapter is partly based on Amersbach, C.; Winner, H.: Functional Decomposition to Reduce Approval Effort for HAD (2017).

[195] Lotz, F. G.: Diss., Referenzarchitektur für die Fahrzeugführung (2017).

[196] Matthaei, R.: Diss., Wahrnehmungsgestützte Lokalisierung (2015).

[197] Reschka, A.: Diss., Fertigkeiten-und Fähigkeitengraphen (2017).

The *sense-plan-act* scheme fulfills the requirement of independence as well as the requirement of generality as all system architectures can be mapped to this high-level decomposition. However, when decomposing the system in only three layers, the possible benefits of particulate testing will be reduced to a minimum.

From the different existing decomposition schemes of the human driving task, that are summarized in section 2.2.2.1, the models by Rasmussen[198] and Zimmer[199], as well as by Reason[200] and Hacker[201] are focusing on error classification rather than decomposition of the driving task and therefore cannot be directly used for the intended decomposition.

The combination of the three-layer models by Donges[202] and Rasmussen[203] is human-focused and not directly transferable to automated driving functions without adaption.

The remaining models by Endsley[204], Wickens[205] and Graab et al.[206a] are quite similar to each other. All of them use the functional components *(Behavioral) Decision* and *(Performance of) Action(s)* that are equivalent to the layers *plan* and *act* of the *sense-plan-act* scheme. The *sense* layer is further decomposed differently in all of the three schemes. As the three schemes fulfill the requirements of independent layers and generality, they are used as a basis for the 6-layer decomposition for HAD functions that is proposed in this work. The six layers are defined in detail in the following sections, their interfaces are described in detail in section 4.2.

### 4.1.1 Layer 0: Information Access

This basic layer is equivalent to layer 1 of the decomposition by Graab et al. and corresponds to "*Stimuli*" in Wickens' model and to "*State of the Environment*" in Endsley's model. It is mainly influenced by the infrastructure, weather, and objects. It is applicable for all kinds of driving functions and all levels of automation. It describes which information is generally accessible. According to Graab et al.[206b] faults on layer 0 could be non-accessible information, obstructed information or masked information due to weather effects (e.g. snow-

[198] Rasmussen, J.: Human errors (1982).

[199] Zimmer, A.: Wie intelligent darf/muss ein Auto sein? (2001).

[200] Reason, J.: The Contribution of Latent Human Failures to the Breakdown of Complex Systems (1990).

[201] Hacker, W.: Allgemeine Arbeitspsychologie (1998).

[202] Donges, E.: Aspekte der aktiven Sicherheit bei der Führung von Personenkraftwagen (1982).

[203] Rasmussen, J.: Skills, rules, and knowledge […] in human performance models (1983).

[204] Endsley, M. R.: Toward a theory of situation awareness in dynamic systems (1995).

[205] Wickens, C. D.: Engineering psychology and human performance (1992).

[206] Graab, B. et al.: Analyse von Verkehrsunfällen […] für die Entwicklung adaptiver FAS (2008).a: -; b: p.9.

covered lane markings). Adapted to automated driving functions, missing or defective information from a digital infrastructure (i.e. digital map, vehicle-to-everything communication (V2X)) are also included in layer 0. The mounting positions of the environment perception sensors have an influence on their field of view, which has an influence on the information access when it comes to obstructed information. It can be assumed that the best achievable mounting position has already been chosen before functional testing. Nevertheless, to ensure safe functionality of the HAD function, the information access has to be taken into account even if it is not an active part of the HAD function itself as failures in layer 0 have to be detected and compensated by the OUT.

### 4.1.2 Layer 1: Information Reception

Layer 1 in general corresponds to layer 2 from Graab et al., layer 1 from Endsley and one part of the perception block in Wickens' model. The information reception layer contains all environment perception sensors of the OUT as well as V2X or backend communication channels and all sensors that are required for the self-representation of the OUT. Advanced information processing - e.g. object detection and tracking - is not part of layer 1. However, the information perception of human drivers is different to machine perception. While the majority of failure modes in human perception are related to distraction or missing attention,[207] those failure modes are unknown for automated vehicles. Nevertheless, there are other failure modes in machine perception that are relevant. Exemplary failures to occur in layer 1 would be a dirty camera that cannot receive all accessible information or limited V2X reception e.g. in a tunnel. The output of layer 1 is sensor raw data and data received via V2X communication as well as data from internal sensors.

### 4.1.3 Layer 2: Information Processing

Layer 2 correspondents to level 2 from Endsley, one part of layer 3 from Graab et al. and a part of the perception block from Wickens. Sensor fusion, object classification, and the generation of an environment model are contained in the information processing layer. Typical failures in this layer would be object existence or state uncertainties or object classification errors.[208] The output of layer 2 is a subjective scene representation from the OUTs point of view according to Ulbrich et al.[209, 210]

---

[207] Graab, B. et al.: Analyse von Verkehrsunfällen […] für die Entwicklung adaptiver FAS (2008), p. 9.

[208] cp. Dietmayer, K. C. et al.: Representation of fused environment data (2016).

[209] Ulbrich, S. et al.: Defining […] the terms scene, situation, and scenario for automated driving (2015).

[210] Cp. section 2.1.1.2.1.

### 4.1.4 Layer 3: Situational Understanding

This layer corresponds to level 3 from Endsley, one part of layer 3 from Graab et al. and a part of the perception block from Wickens. It is derived from layer 2's scene representation by goal- and value-specific information selection (i.e. elements from the scene that are not relevant for the mission of the OUT are filtered out) and augmentation (i.e. semantic information is added). The output of layer 3 is a situation representation according to Ulbrich et al.[211, 212] It contains all information and constraints that are required for the behavioral decision in layer 4. Failures like wrong predicted trajectories of object vehicles can occur in layer 3.

### 4.1.5 Layer 4: Behavioral Decision

This layer corresponds to layer 4 from Graab et al. and the blocks (*Behavioral) Decision* from Endsley and Wickens and thus represents the classical *plan* layer. It contains the algorithms that decide - based on the situation model and behavioral competences and mission goals - about the behavior of the HAD function. In contrary to human drivers, it is not expected that automated driving functions deliberately violate the rules. Therefore, the only failure types that can occur in layer 4 of an automated driving function are decision or planning mistakes. An exemplary failure would be an error in the maneuver planning that leads to a collision with another vehicle. The output of layer 4 is the planned trajectory.

### 4.1.6 Layer 5: Action

The final layer is equivalent to the final layers in the underlying models. It transforms the trajectory from layer 4 into the actual vehicle movement. It includes the vehicle's motion control algorithms as well as the necessary actuators. In contrary to human drivers, the failure modes are different for automated driving functions. Instead of operation errors (e.g. confusion of accelerator and brake pedal) or wrong reaction (e.g. inadequate steering input), a typical failure of an automated driving function in layer 5 would be an unstable or offset-affected motion control algorithm.

---

[211] Ulbrich, S. et al.: Defining […] the terms scene, situation, and scenario for automated driving (2015).

[212] Cp. section 2.1.1.2.1.

## 4.2 Definition of Generic and Observable Interfaces

In this subchapter, research question 2 is addressed:

Q 2: *How can the interfaces between the functional layers derived in Q1 be defined generically?*

In order to evaluate the functional layers on their interfaces, those interfaces have to be defined. This definition has to be generic, as the method should be applicable to all automated driving functions. Furthermore, the interfaces have to be accessible in order to feed input data in and observable in order to evaluate the output. Where it is feasible, already existing or even standardized descriptions and data formats are used for the interfaces. In the following sections, the interfaces are defined in detail.

### 4.2.1 Input to Layer 0: Existing Information

The input to layer 0 is all existing information[213] in the test case. In virtual environments, the standardized interface *Open Simulation Interface (OSI)*[214] is recommended, as its *"*[…] *GroundTruth interface gives an exact view on the simulated objects in a global coordinate system."*[214]. It *"*[…] *is supposed to describe the whole simulated environment around any simulated vehicle"*[215]. In real test environments, the existing information has to be created with suitable reference sensors and digital map data. In addition to the existing information of the scene, general information such as traffic rules within the ODD and the self-representation of the OUT have to be included in this interface.

### 4.2.2 Interface between Layer 0 and Layer 1: Accessible Information

The interface between the layers 0 and 1 is all accessible information, i.e. all information that would be accessible for an ideal HAV or a human. The accessible information is the existing information, excluding all occluded elements, e.g. lane markings or traffic signs that are occluded by dynamic objects or snow. Thus, the interface corresponds to the *SensorView*

[213] The term information in this context is defined as all data that is relevant for conducting the driving task. This includes - but is not limited to - states of dynamic objects, ego and traffic lights, street layout and applicable traffic rules as well as the current system capability's that might be restricted by influences from the environment or system degradation.

[214] Hanke, T. et al.: OSI: A generic interface for the environment perception of AD functions (2017).

[215] Cp. GitHub: open_simulation_interface: osi3::GroundTruth Struct Reference (2019).

*Struct*[216] in OSI. In contrary to the existing information, this interface is specific for the OUT, as sensor-mounting positions have an influence on occlusion of information. Additionally, accessible V2X information has to be included in the interface as well. Currently, there is no interface for V2X specified in OSI. Therefore, it is recommended to extend OSI with an interface for V2X information according to existing standards as IEEE 802.11p or C-V2X.[217] In real test environments, the accessible information has to be generated similar to existing information with reference sensors.

### 4.2.3 Interface between Layer 1 and Layer 2: Sensor Raw Data

The interface between the layers 1 and 2 are the actually received information, without advanced post-processing (e.g. object detection) other than sensor-internal processing steps such as demodulation or digitalization of analog data. Feasible representations of those sensor raw data would be for example point clouds for lidar or radar reflections. Standards for such sensor raw data interfaces are currently under development in the ISO working group *ISO/TC 022/SC 31/WG 09 "Sensor data interface for automated driving functions"* [218] and are planned to be published in 2020.[219] However, the interface cannot be used with many state-of-the-art hardware sensors, as raw data interfaces are physically not available due to sensor-internal data processing. Additionally, the system's self-representation has to be included in this interface.

### 4.2.4 Interface between Layer 2 and Layer 3: Subjective Scene Representation

As stated already in section 4.1.3, the interface between the layers 2 and 3 is a subjective scene representation from the OUT's perspective according to Ulbrich et al.[220]. While objective scenes can be represented in the standardized formats OpenDRIVE®[221] and OpenSCENARIO®[222], there is - to the author's best knowledge - no standard format for subjective scene representations available, yet. Defining a standard interface for a subjective scene representation will be a challenge, as various types for environment representations

[216] GitHub: open_simulation_interface: osi3::SensorView Struct Reference (2019).

[217] Cp. Vukadinovic, V. et al.: 3GPP C-V2X and IEEE 802.11 p for V2V communications (2018).

[218] ISO: ISO/TC 22/SC 31 - Data communication.

[219] Schaller, T.; Dehlink, B.: Sensor standardization initiative for automated driving (2017).

[220] Ulbrich, S. et al.: Defining […] the terms scene, situation, and scenario for automated driving (2015).

[221] VIRES GmbH: OpenDRIVE® / OpenCRG® Product Data Sheet (2011).

[222] VIRES GmbH: OpenSCENARIO Homepage (2018).

(e.g. grid-based, object-based, hybrid) exist and different coordinate systems are used. [223] Thus, it might be necessary to define standardized formats for each of the environment representation types and use the format that is appropriate for the OUT.

### 4.2.5 Interface between Layer 3 and Layer 4: Situation Representation

The interface between the layers 3 and 4 is a situation representation while the content of the interface is defined in detail by Ulbrich et al.[220], similar to the subjective scene representation there is no standardized data format available to represent a situation. Due to the reasons described in section 4.2.4 it might not be feasible to define one standard format that is suitable for all types of situation representations.

### 4.2.6 Interface between Layer 4 and Layer 5: Planned Trajectory

The interface between layer 4 and 5 is the planned trajectory. A trajectory consists of a path, i.e. geometry information, and a time component.[224] There exist different options for path representation (e.g. splines, polynomials, discrete waypoints, etc.) and the time component can be implicit in form of a velocity profile or explicit in form of a timestamp for each waypoint. Thus, in combination, there are multiple options for trajectory representation. However, they can be transformed into each other to some extent. Thus, any kind of trajectory representation can be used for the interface, as long as a converter to the trajectory representation that is required by the test evaluation tool exists.

### 4.2.7 Output from Layer 5: Observable Behavior

The output of layer 5 is the observable behavior of the OUT. This is mainly the vehicle's movement, which is supplemented by other actions, such as operating the indicator. For an automated evaluation of the vehicle's movement, it is recommended to use the driven trajectory. Therefore, in real test environments it is required to record the driven trajectory with adequate measurement devices. Whereas validated vehicle models are required in simulation. Similar to the planned trajectory, any kind of trajectory representation can be used for the interface, as long as a converter to the trajectory representation that is required by the test evaluation tool exists.

---

[223] Cp. Dietmayer, K. C. et al.: Representation of fused environment data (2016).

[224] Rathgeber, C.: Diss., Trajektorienplanung und-folgeregelung (2016), p. 31.

## 4.3 Summary and Exemplary Application of the Proposed Decomposition Scheme

Figure 4-1 summarizes the proposed functional layers for decomposition and the corresponding interfaces. For some of the interfaces, standardized specifications exist already or are in development. For other interfaces, standards have yet to be developed for an efficient application of the method. However, if one interface does not exist or is not accessible, which is for example the case for sensor raw data interfaces in current series applications of ADAS, the corresponding functional layers have to be tested and evaluated in combination.

If functional decomposition and particulate testing are intended to be used for the validation of new systems, the interfaces should be provided in system architecture design.

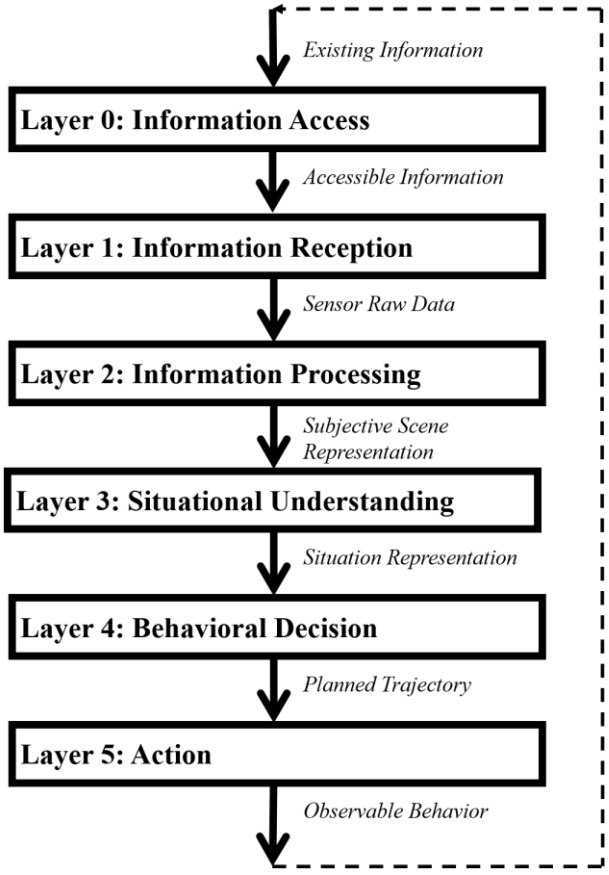


Figure 4-1: Functional layers and interfaces.

The derived decomposition scheme is exemplarily applied to the system architectures from Lotz[225] (see Figure 4-2) and Matthaei[226] (see Figure 4-3). It becomes obvious that layer 0 is

[225] Lotz, F. G.: Diss., Referenzarchitektur für die Fahrzeugführung (2017).

[226] Matthaei, R.: Diss., Wahrnehmungsgestützte Lokalisierung (2015).

not covered by both architectures, as the sensor mounting positions are not included in functional system architectures. Furthermore, the *HMI* respective *Communication* blocks are not covered by the decomposition scheme as they are outside the scope of this work (cp. section 3.1.1).

In general, the decomposition scheme is applicable to both system architectures, even though the boundary between the layers 2 and 3 in Matthaei's architecture is vague.

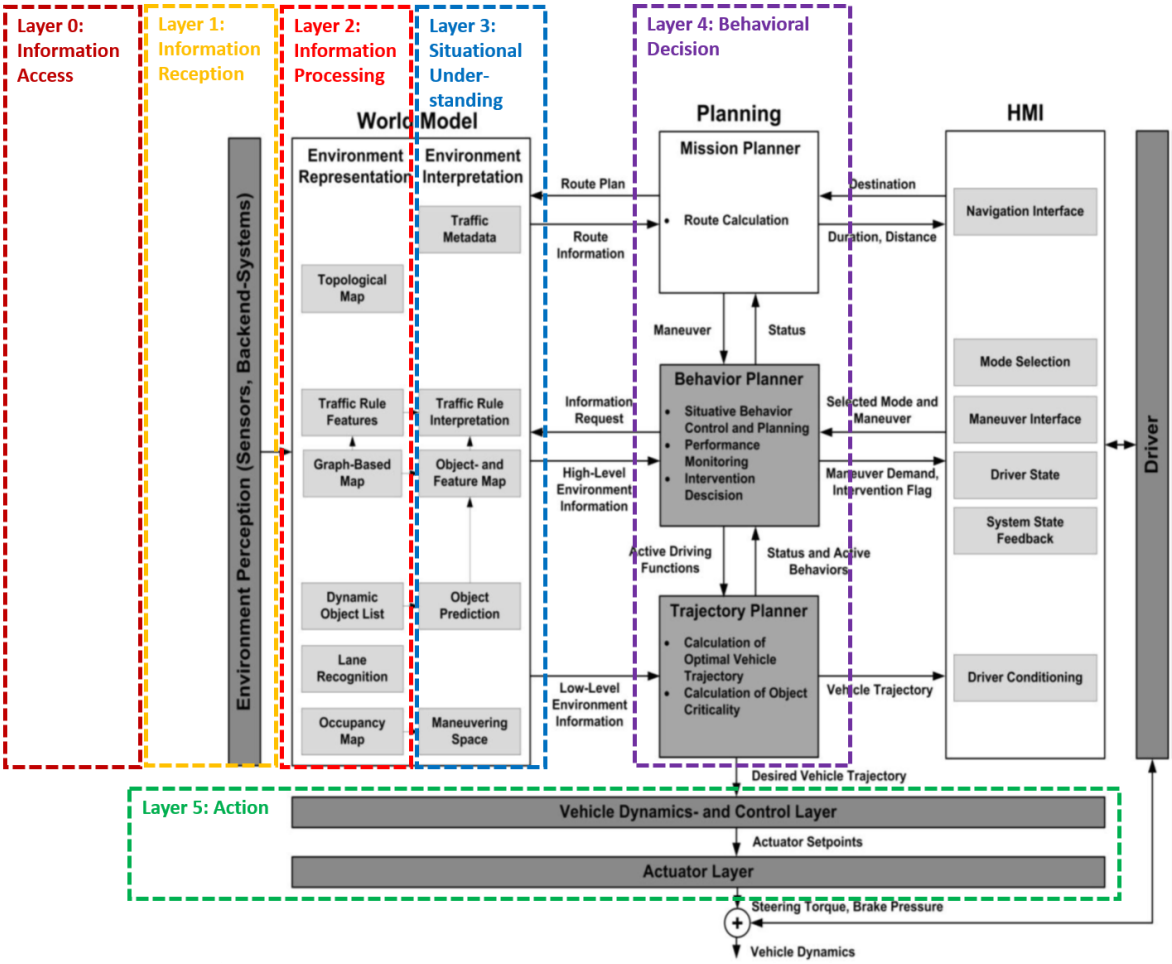


Figure 4-2: Decomposition scheme applied to the system architecture by Lotz.[227]

[227] Based on Hohm, A. et al.: Automated driving in real traffic (2014), p. 11.

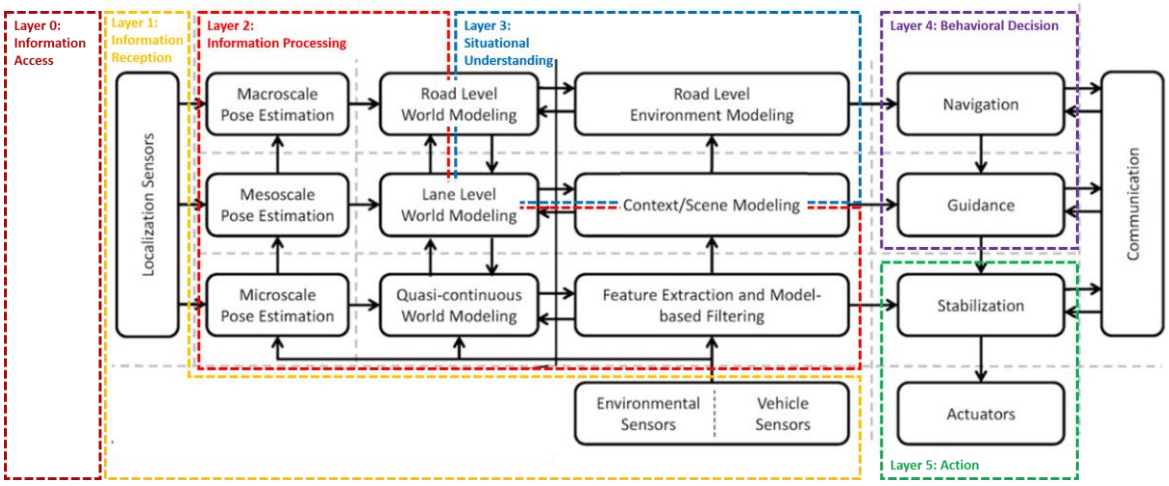


Figure 4-3: Decomposition scheme applied to the system architecture by Matthaei.[228]

## 4.4 Interim Conclusion

A 6-layer functional decomposition scheme for HAD functions that is based on existing decomposition schemes for the human driving task is proposed. Corresponding interfaces between the single functional layers are defined. For some of the interfaces there are already standards - at least for virtual environments - available that can be adapted to be used for the decomposition approach. For other interfaces, e.g. for sensor raw data, standards are in development now. However, they cannot be used with state-of-the-art hardware yet, as physical interfaces are not available (except for dedicated development sensors). For the remaining interfaces, the content has been defined while standardized data formats still have to be developed. Mapping the proposed decomposition scheme to two generic system architectures for HAD functions demonstrates that it is applicable to different system architectures.

[228] Based on Matthaei, R.; Maurer, M.: Functional System Architecture (2018), p. 97.

# 5 Particulate Evaluation Criteria[229]

In this chapter, research question 3 is addressed:

Q 3: *How can particulate evaluation criteria for decomposed HAD functions be derived?*

As already outlined in section 3.1.2, the evaluation criteria on system level are prerequisites for the method. Therefore, the definition of system level evaluation criteria is neither included in the method nor in this work. However, an overview of the state of the art for the derivation of evaluation criteria is given in section 2.1.3.

While at least some of the system-level evaluation criteria are straightforward, the derivation of particulate evaluation criteria for the single functional layers is challenging. For example, an obvious fail criterion on system level is a collision with another traffic participant. In this case, the evaluation on system level is solely based on the OUT's observable behavior. However, there are multiple reasons in the different functional layers that can cause faulty behavior on system level. In the above-mentioned example, possible reasons for the collision could be faults in the information reception or information processing that lead to false negative detections or wrong object state representation. A wrong interpretation of the object's intention in layer 3, as well as a decision mistake (e.g. braking instead of evasion) in layer 4 or a deviation from the planned trajectory in layer 5 could be other reasons.

In the following subchapters, a methodology to derive particulate evaluation criteria is proposed and examples for the application of system-theoretic process analysis (STPA)[230] and fault tree analysis (FTA) to derive evaluation criteria are given. Eventually, an approach that could be used for traceable modeling of the dependencies between system-level evaluation criteria and particulate evaluation criteria is introduced.

## 5.1 Methodology to Derive Particulate Evaluation Criteria

Breaking down evaluation criteria from system level to functional layer requires an adequate and systematic method to avoid mistakes and to maintain and generate as much relevant information about evaluation criteria as possible.

[229] Some parts of this chapter have been published already in Klamann, B. et al.: Defining Pass-/Fail-Criteria (2019). And Amersbach, C.; Winner, H.: Functional Decomposition to Reduce Approval Effort for HAD (2017).

[230] Leveson, N.: An STPA Primer (2013).

Therefore, as a first step of a methodical derivation of particulate evaluation criteria, it is recommended to use established safety analysis methods such as FTA and STPA to derive causal factors from evaluation criteria or safety goals on system level. The causal factors are possible causes for violation of safety goals or evaluation criteria. Other methods such as common cause analysis (CCA) or root cause analysis (RCA) should be used in addition to guarantee completeness of the list of causal factors. In a second step, the causal factors are allocated to functional layers. Thereafter, particulate evaluation criteria are derived from the allocated causal factors for each functional layer. The methodology for the derivation of particulate evaluation criteria is summarized in Figure 1-1.

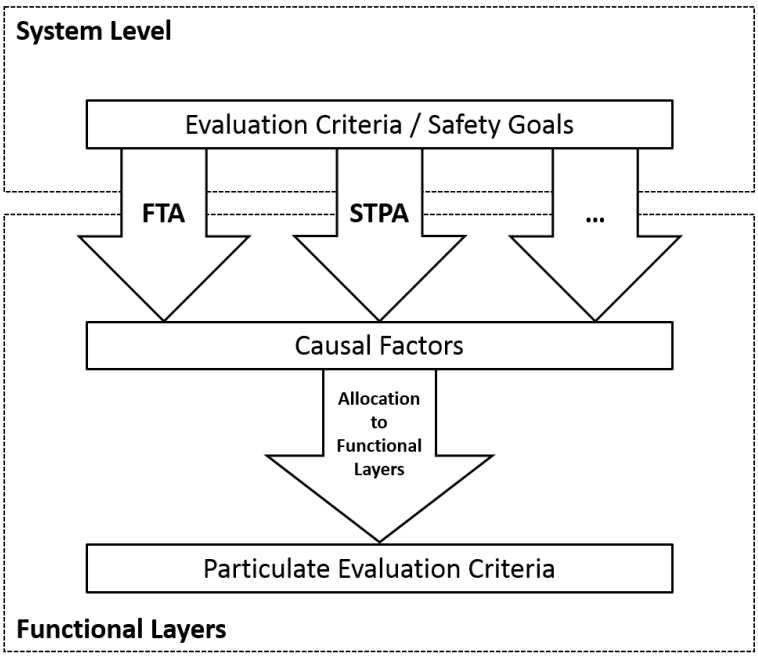


Figure 5-1: Methodology to derive particulate evaluation criteria.

According to Schuldt et al.[231], the evaluation of test cases can only be efficient if it is done automatedly. In order to allow for an automated evaluation, all evaluation criteria have to be formulated in a computable way. Therefore, they have to be formulated for example as mathematical equations including distinct threshold values. Furthermore, the evaluation criteria can be hard or soft. While the violation of a hard evaluation criterion is K.O. criterion that directly leads to an evaluation of the test case with "failed", the soft evaluation criteria can be used for a more detailed evaluation and do not necessarily lead to a "fail" evaluation. An example of a K.O. criterion could be a collision with another traffic participant, while lateral acceleration above a comfort-related threshold would be a soft criterion.[231]

[231] Schuldt, F. et al.: Effiziente systematische Testgenerierung für FAS in virt. Umgebungen (2013).

## 5.2 Exemplary Derivation of Evaluation Criteria Using FTA

For an exemplary derivation of particulate evaluation criteria via FTA, the functional scenario that led to a real-world accident of a Tesla Model S in Switzerland in May 2016[232] is used as an example. The scenario is visualized in figure 2:

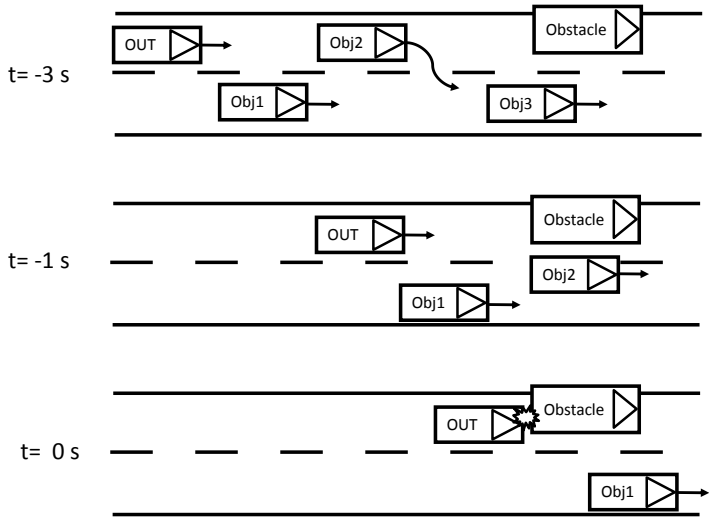


Figure 5-2: "Swiss scenario".[233]

Initially, the OUT is driving in the left lane of a two-lane highway in dense traffic. A van (Obstacle) is standing on the left side of the left lane. The lead vehicle in front of the OUT (Obj2) then performs a lane change to the right lane to avoid the obstacle while the OUT does not attempt to avoid the obstacle and crashes into it. In this example, it is assumed that the OUT is a level 3 vehicle (according to SAE J3016[234]) instead of a level 2 vehicle as in the real accident.

The accident is used as a top event for an FTA to derive fail criteria for this functional scenario. Figure 5-3 shows an extract from the exemplary failure tree. Starting with the collision as the top event, the causal factors would be a failure in the collision avoidance or that a collision avoidance is not planned at all. Following the branch of the faulty collision avoidance, one of the base events that could lead to a collision over several intermediate steps is that the sensor range is lower than the physically required braking distance. This would be

[232] Lambert, F.: Tesla Model S driver crashes into a van while on Autopilot [Video] (2016).

[233] Amersbach, C.; Winner, H.: Functional Decomposition to Reduce Approval Effort for HAD (2017).

[234] SAE: J3016: Taxonomy and Definitions for […] Vehicle Automated Driving Systems (2018).

allocated as evaluation criteria to layer 1 after formulating it in a computable way and defining threshold values that are in this example dependent on external parameters, e.g. the coefficient of friction.

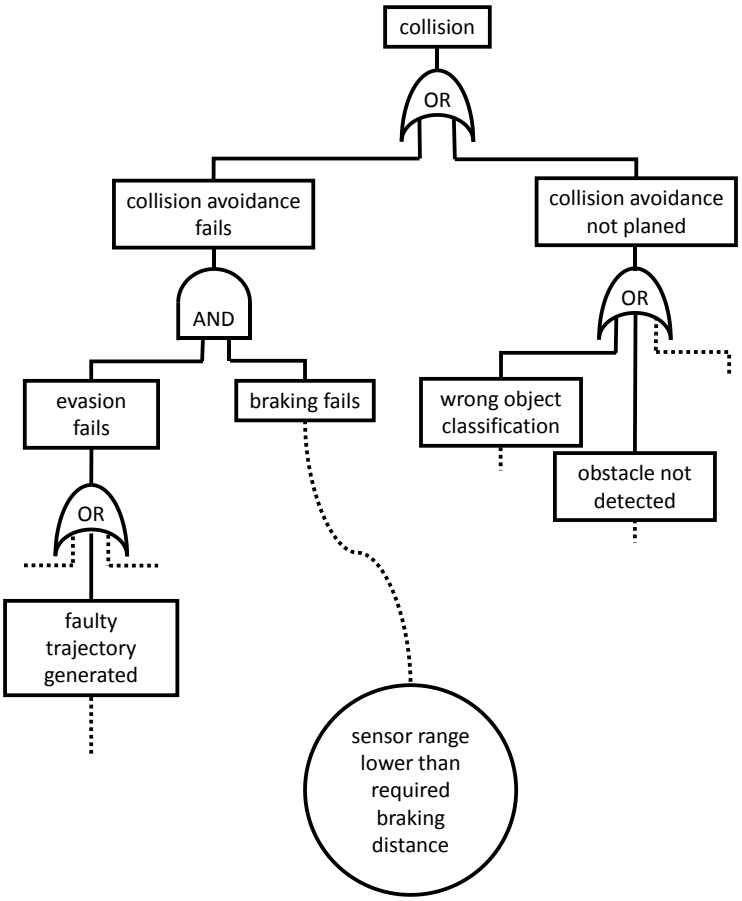


Figure 5-3: Simplified exemplary failure tree.[235]

This example serves to illustrate the procedure in general and therefore uses a lot of simplifications and is limited to high-level error causes. The fault tree for a real system would be much more complex and would contain more possible causal effects, e.g. errors in the communication between two layers or between components within a layer.

## 5.3 Exemplary Derivation of Evaluation Criteria Using STPA

As an example for the application of STPA, the method is used to derive and allocate causal factors from the potential safety goal (SG) *"Overrunning solid lines must be prevented"*

[235] Amersbach, C.; Winner, H.: Functional Decomposition to Reduce Approval Effort for HAD (2017).

which is adapted from SG 12 in Stolte et al.[236] As a first step for STPA, the OUT is modeled as a control circuit. The control circuit for a decomposed HAD function is shown in Figure 5-4. It shows the functional layers as boxes inside and outside the HAV and their defined interfaces.

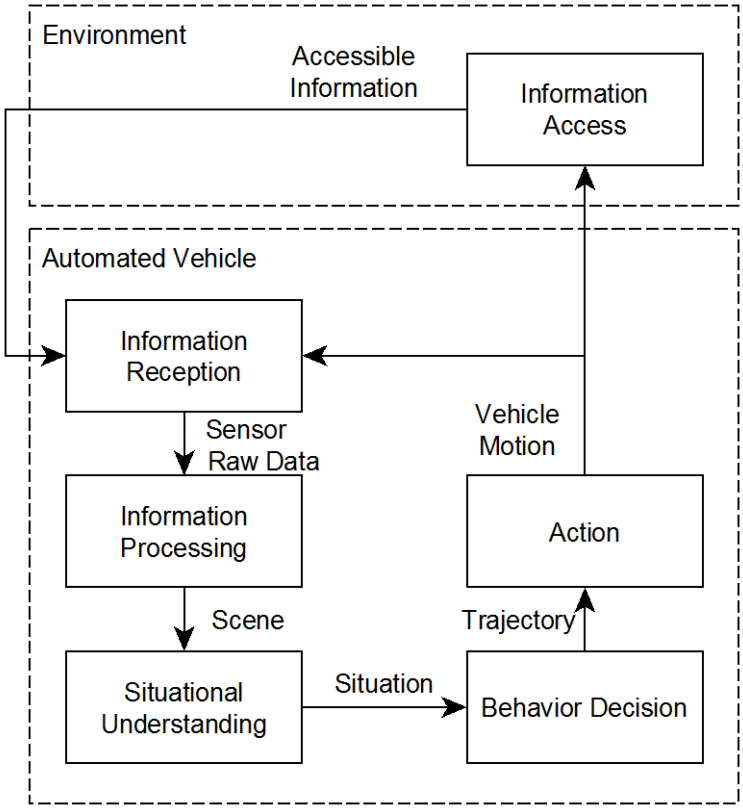


Figure 5-4: Control Circuit for STPA of a decomposed HAD function.[237]

In STPA, the interfaces are analyzed regarding potential unsafe (control) actions for four basic classes: [238]

1. *not providing a control action causes a hazard*
2. *providing a control action causes a hazard*
3. *providing a control action too late, too early or in wrong order causes a hazard*
4. *providing a control action too long or stopping it too early causes a hazard*

Figure 5-5 shows an example for the first two categories of unsafe actions analyzed in STPA. Classes 3 and 4 are not included in this simplified example.

The output of the functional layer *information reception* causes the two given unsafe control actions. One causal factor for the unsafe control action "*no processible sensor raw data of solid line markings*" is that a sensor is covered. Safety requirements to either prevent a sensor

[236] Stolte, T. et al.: Hazard analysis and risk assessment for an automated unmanned protective vehicle (2017).

[237] Klamann, B. et al.: Defining Pass-/Fail-Criteria (2019).

[238] Leveson, N.: An STPA Primer (2013), p. 18.

from being covered or discover a covered sensor are defined in the following step. However, the safety requirements to discover causal factors that lead to unsafe behavior will not be sufficient and demand further safety requirements. Even though, the process to define pass /fail criteria does not change and is done in the first step by simply inverting the safety requirement. This leads to the fail-criteria that a "*sensor is covered during operation*" OR that a "*covered sensor is not detected during operation*".

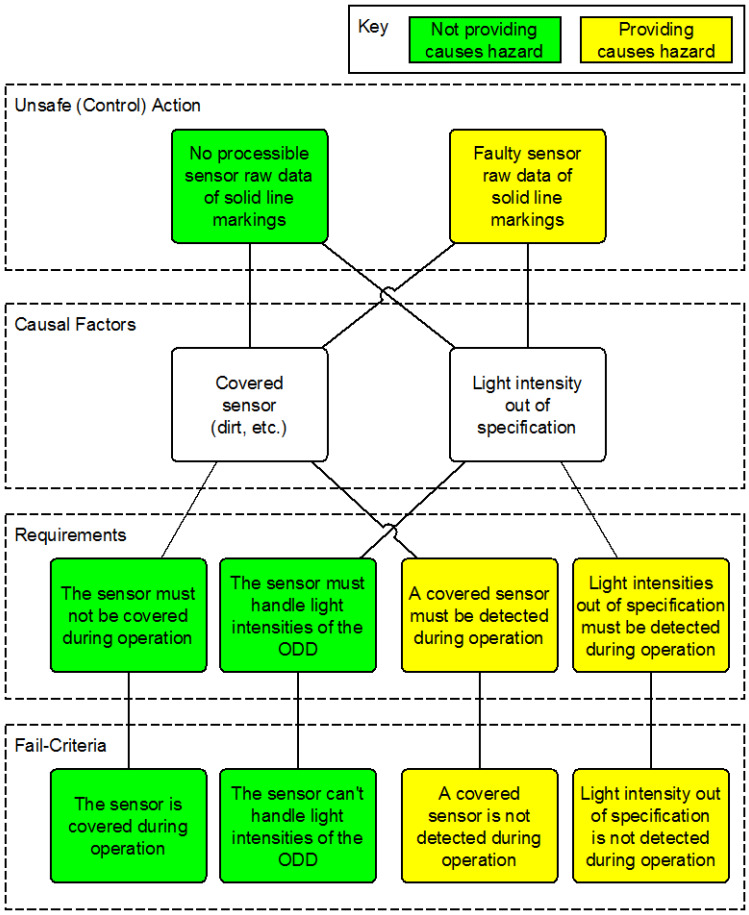


Figure 5-5: Exemplary derivation of particulate evaluation criteria using STPA.[237]

The developed causal factors are still defined qualitatively by terms such as "wrong" or "failure". To convert the derived pass/fail criteria to testable criteria, metrics and dedicated thresholds must be defined. Especially for virtual testing with automated evaluation, it is important to refer to defined values or ranges of values. Therefore, for example, "coverage" has to be quantified by a maximum area of masking while the definition of masking might be a specific light permeability.

A more detailed example for the application of STPA to derive safety requirements is given by Stolte et al.[239].

[239] Stolte, T. et al.: Safety goals and functional safety requirements (2016).

## 5.4 Automatic Derivation of Dependency Chains

Hoßbach[240a] proposes an approach for the automatic derivation of dependency chains. A *chain derivation engine* that includes schematic rules is used to derive dependency chains that are intended to support tractability of system requirements and design decisions. As part of the approach, *task-chain pattern skill graphs* that are based on the *skill graphs* by Reschka[241] are used to model the system architecture. *"They represent a system's architecture as the union of a multitude of skills and their dependencies on each other."* [240b] An exemplary task-chain pattern skill graph can be seen in Figure 5-7.

Based on the task-chain pattern skill graphs and ontology-based representations of relevant scenarios, dependency chains between system requirements and component requirements are derived automatically. An exemplary dependency chain is shown in Figure 5-6.

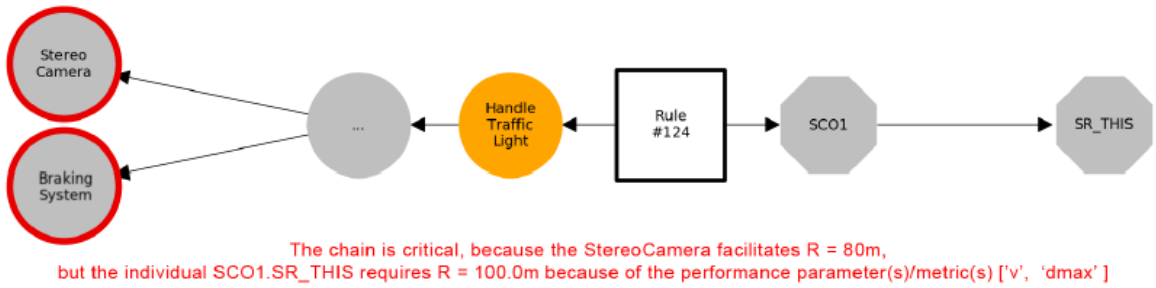


Figure 5-6: Exemplary dependency chain. [240c]

The approach proposed by Hoßbach could be adapted for traceable modeling of the dependencies between system-level evaluation criteria and particulate evaluation criteria and thus enable the automated derivation of particulate evaluation criteria.

## 5.5 Interim Conclusion

Common methods from safety engineering such as FTA and STPA are used as a basis to derive causal factors on the individual functional layers from evaluation criteria or safety goals on system level. To ensure completeness, they can be supplemented by additional safety analysis methods such as CCA and RCA. From those causal factors, particulate evaluation criteria are derived. For an automated test evaluation, those criteria have to be formulated in a machine-readable way including threshold values. An automated dependency-chain derivation approach that is based on ontologies and rule engines can be used as a basis for the automated derivation of evaluation criteria.

---

[240] Hoßbach, P. M.: Masterthesis, Automatic Derivation of Dependency Chains (2019). a: -; b: p. 47; c: p. 48.

[241] Reschka, A.: Diss., Fertigkeiten-und Fähigkeitengraphen (2017).

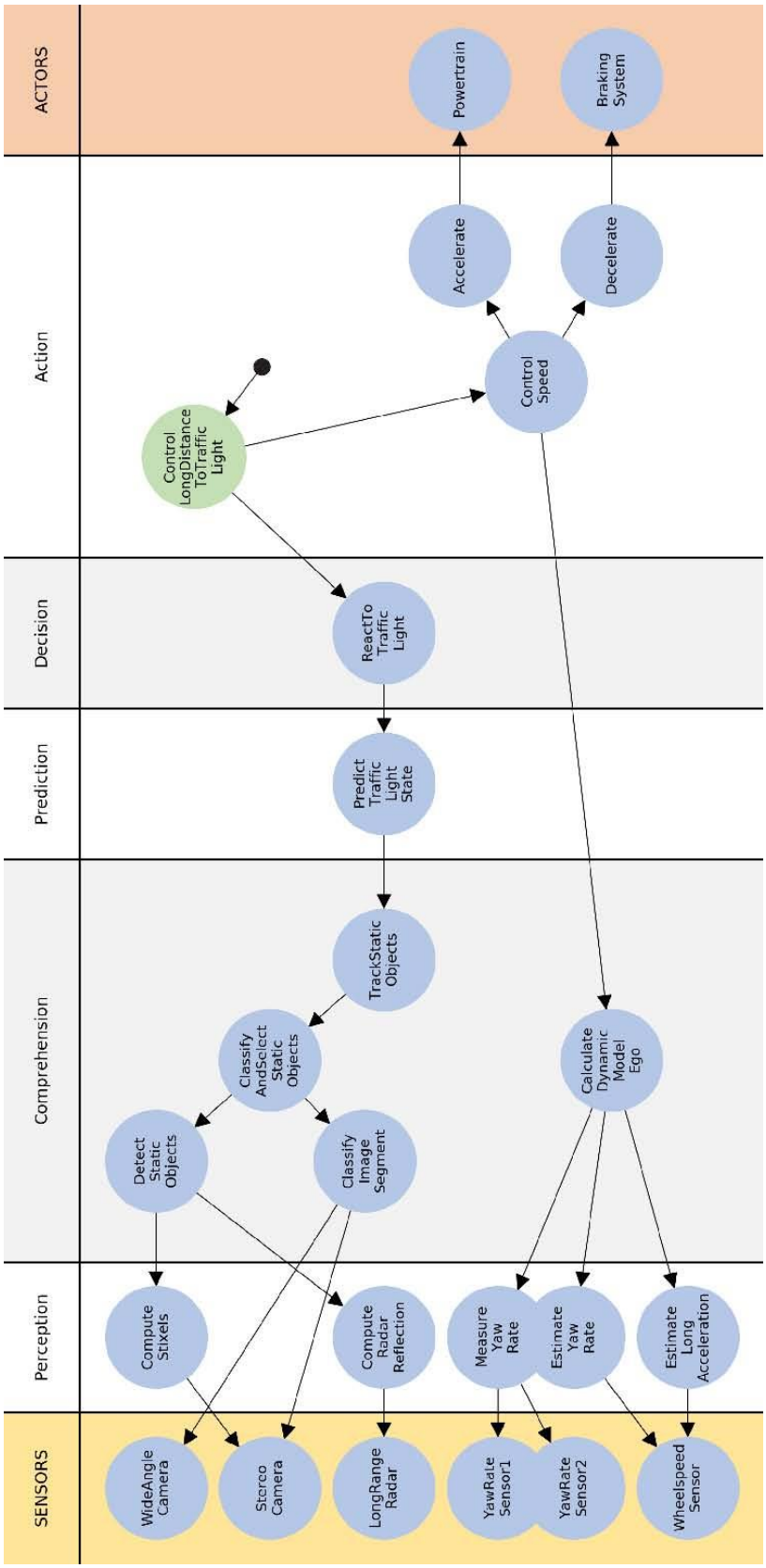


Figure 5-7: Exemplary task-chain pattern skill graph.[242]

[242] Hoßbach, P. M.: Masterthesis, Automatic Derivation of Dependency Chains (2019), p. 56.

# 6 Allocation of Influence Parameters to Functional Layers

This chapter deals with research question four:

Q 4: *How can influence parameters be allocated to the functional layers derived in Q1?*

To answer this research question, first the influence parameters on system level have to be selected and analyzed. Thereafter, the external influence parameters can be allocated to functional layers and transformed to input parameters on the interfaces of the functional layers. Additional internal influence parameters have to be considered.

## 6.1 Selection and Analysis of External Influence Parameters on System Level

Schuldt[243] proposes a method to select and analyze external influence parameters on system level. As requirements, he defines that the selection has to identify parameters that have an influence on the OUT as well as representative value ranges for those parameters. Additionally, he states that the analysis has to define discretization steps for continuous parameters as well as the error probability through discretization.

According to Schuldt[243], the following information sources are available for the definition of influence parameters:

- Derivation from the functional specification and therein-contained operation scenarios
- 6-layer-model for the description of scenarios[244]
- Standards and guidelines for the construction of roads[245]
- Derivation from driving maneuvers
- Derivation from accident analysis
- Expert knowledge

[243] Schuldt, F.: Diss., Ein Beitrag für den methodischen Test von autom. Fahrfunktionen (2017), pp. 85–102.

[244] Schuldt is using his original 4-layer-model that was extended to 6-layers by Bagschick et al. (2018) and Sauerbier et al. (2019); compare section 2.1.1.2.2 and Figure 2-2.

[245] e.g. FGSV: Richtlinien für die Anlage von Autobahnen (RAA) (2008).

For the analysis of influence parameters, Schuldt[246] describes the following information sources and approaches:

- Optimization algorithms
- Statistical evaluation
- Expert knowledge

The selection and analysis of influence parameters on system level according to Schuldt is summarized in Figure 6-1. As those steps are a pre-requirement and not a component of the functional decomposition methodology developed in this work, they are not handled in detail here, but referenced to Schuldt[246a] instead.

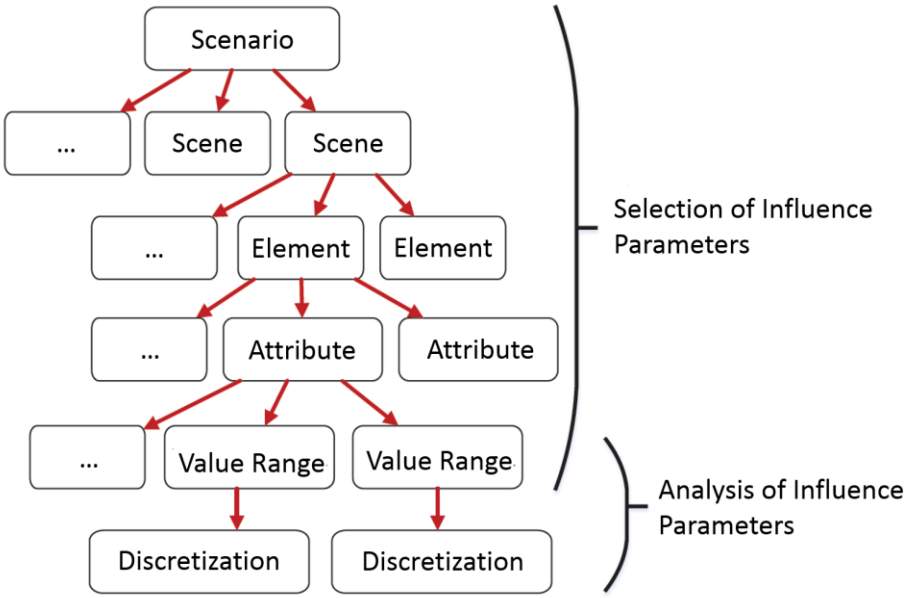


Figure 6-1: Selection and analysis of influence parameters according to Schuldt.[246b]

## 6.2 Allocation of External Influence Parameters to Functional Layers

One of the main motivations for the functional decomposition approach is the fact that the majority of influence factors is only relevant to one or a few functional layers (cp. subchapter 3.1 and section 8.3.2). To use this fact for the reduction of validation effort, the external influence parameters have to be allocated to the functional layers.

[246] Schuldt, F.: Diss., Ein Beitrag für den methodischen Test von autom. Fahrfunktionen (2017).a: pp. 85–102; b: translated from p. 97.

### 6.2.1 Intuitive Allocation of Influence Parameters

For small scenario catalogs with a limited parameter space, the allocation of influence parameters to functional layers can be done intuitively by experts. This method has been applied for the exemplary scenario set that has been used for the analysis in subchapter 8.3 (see Appendix A.1). However, as the intuitive allocation of influence parameters is based on expert knowledge, it is not reproducible and thus violates one of the requirements derived in section 3.1.4. Additionally, expert knowledge is only available for evolutionary development and cannot be used for disruptive development.[247] Finally, the manual, intuitive allocation of influence parameters by experts is not feasible for large-scale applications.

### 6.2.2 Rule-Based Allocation of Influence Parameters

Bickel[248a] proposed a rule-based allocation of external influence parameters to functional layers. The method allocates external influence parameters from the 6-layer model for scenario representation based on simple decision rules. The rules are summarized in a decision tree that can be seen in Figure 6-2.

Comparing the allocation of external influence parameters based on the decision tree with an intuitive allocation, the decision-tree-based allocation leads to a higher number of influence parameters that are allocated to each of the layers. On the one hand, this is partly based on imprecise and generic decision rules that lead to a conservative allocation. On the other hand, while for some parameters the allocation is obvious, for other parameters it depends on a multitude of factors and constraints whether a parameter has an influence on one specific functional layer in a certain scenario or not. Those complex relations cannot be covered by one generic decision tree that covers all scenarios. Therefore, specific decision trees for each functional scenario are recommended by Bickel.[248b]

However, deriving a specific decision tree based on expert knowledge for each scenario is not feasible for big scenario catalogs that are required during the validation of HAV. Therefore, it is proposed to model rules and semantic constraints for the allocation of influence parameters that are based on expert knowledge as well as on systematic analysis in a machine-readable way. This could be done for example by ontologies. A derivation engine similar to the dependency-chain-derivation-engine developed by Hoßbach[249] (cp. subchapter 5.4) can then be used to allocate external influence parameters automatically.

---

[247] Schuldt, F.: Diss., Ein Beitrag für den methodischen Test von autom. Fahrfunktionen (2017), p. 94.

[248] Bickel, J.: Mth., Identifikation, Diskretisierung und Zuordnung von Einflussparametern (2019). a: pp. 68–81; b: pp. 79-81.

[249] Hoßbach, P. M.: Masterthesis, Automatic Derivation of Dependency Chains (2019).

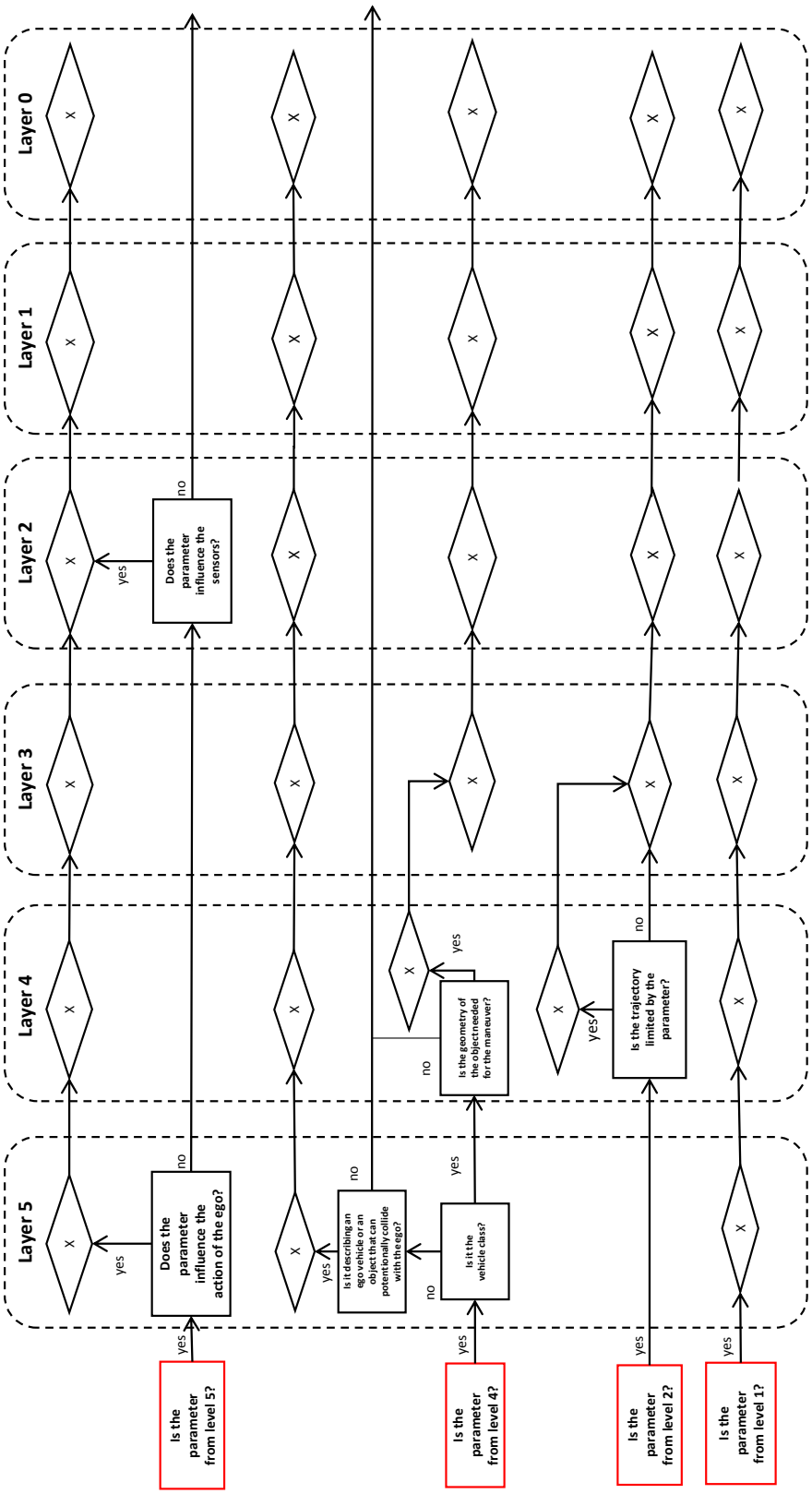


Figure 6-2: Decision tree for the allocation of influence parameters.[250]

[250] Translated from Bickel, J.: Mth., Identifikation, Diskretisierung und Zuordnung von Einflussparametern (2019), p. 71. An „x“ in the decision tree means that the parameter is allocated.

## 6.3 Input Parameters on the Interfaces

The allocation of external influence parameters to functional layers shrinks the relevant parameter space for particulate testing. However, the external parameters cannot be directly used as input for particulate tests of the single functional layers. Schuldt[251a] describes how test harnesses can be used to generate the required input data. According to IEEE[252], a test harness is defined as *"a software module used to invoke a module under test and, often, provide test inputs, control and monitor execution, and report test results."*[253] Hereby it is differentiated between open-loop and closed-loop test harnesses as illustrated in Figure 6-3.

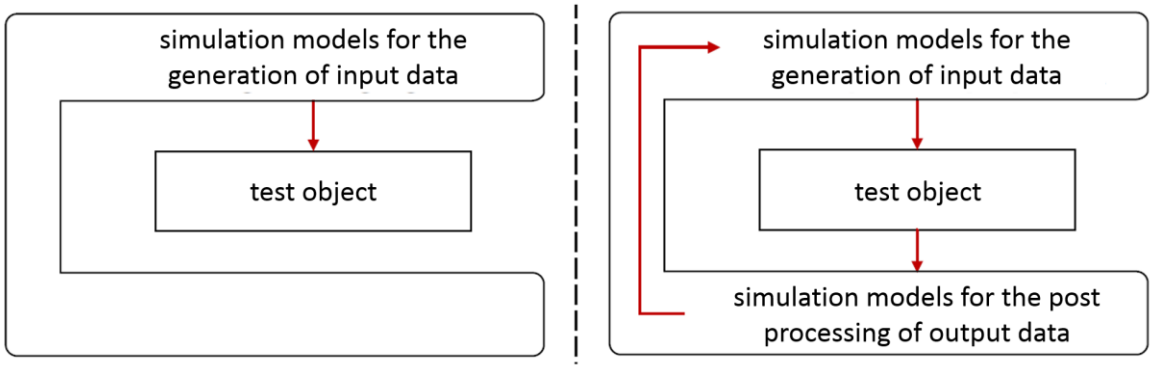


Figure 6-3: Open-loop (left) and closed-loop (right) test harnesses according to Schuldt.[251b]

If the output data of the test object - in this case a functional layer of the automated driving function - have an influence on the input data, closed-loop test harnesses are required. If open-loop test harnesses are sufficient, real measurement data could be used instead of artificially generated input data.

If the test case is carried out in an environment that includes real components, transmitters are required in addition to the simulation models to generate emulated input data, for example for radar or ultra-sonic sensors. [251a]

Depending on the requirements for the test case, the simulation models that are required are already available or still in development. For example, if an ideal scene representation or a scene representation with a simplified error representation is sufficient for a test of layer 3, the input data can be generated with already available ideal sensor models and error generators. Whereas sensor models that can create realistic sensor raw data that might be required as input data for a test of layer 2 are currently not available. In this case, layer 1 and 2 cannot

[251] Schuldt, F.: Diss., Ein Beitrag für den methodischen Test von autom. Fahrfunktionen (2017).a: pp. 144-157; b: translated from p. 145.

[252] IEEE: IEEE 610: Standard Glossary of Software Engineering Terminology (1990), p. 75.

[253] The term „test harness" is consequently used in this work instead of its synonym „test driver" to avoid confusion with the term "test driver" in the sense of a specially trained driver to carry out test drives.

be tested individually as the input data for layer 2 cannot be simulated or emulated. Therefore, a combined test in a suitable XiL environment has to be carried out. However, the evaluation can be done individually for the involved functional layers if the interfaces are observable.

In addition to the transformation of the external influence parameters, the test harnesses have to generate relevant internal influence parameters. Internal influence parameters are parameters from the self-representation of the vehicle such as system states and ego movement. As an example, layer 2 requires information about the ego movement for the sensor data processing. Information on the system state is required in layer 4, as the ego dynamics might be limited if some systems are only working in degraded mode.

## 6.4 Interim Conclusion

As a prerequisite for particulate testing, external influence parameters on system level have to be selected. Several data sources are available for the identification and selection of input parameters. The external influence parameters are allocated to functional layers. For a prototypical exemplary application, this could be done intuitively based on expert knowledge. However, for an application of the functional decomposition approach in the development and sign-off processes, this step has to be automated and objectified. Thereafter, the relevant external influence parameters as well as internal influence parameters and output data from the particulate test are converted to input data for the particulate tests by test harnesses. Depending on the requirements of a specific particulate test, test harnesses are not available for all interfaces in all test environments yet.

The chain for the creation of input parameters is illustrated in Figure 6-4.

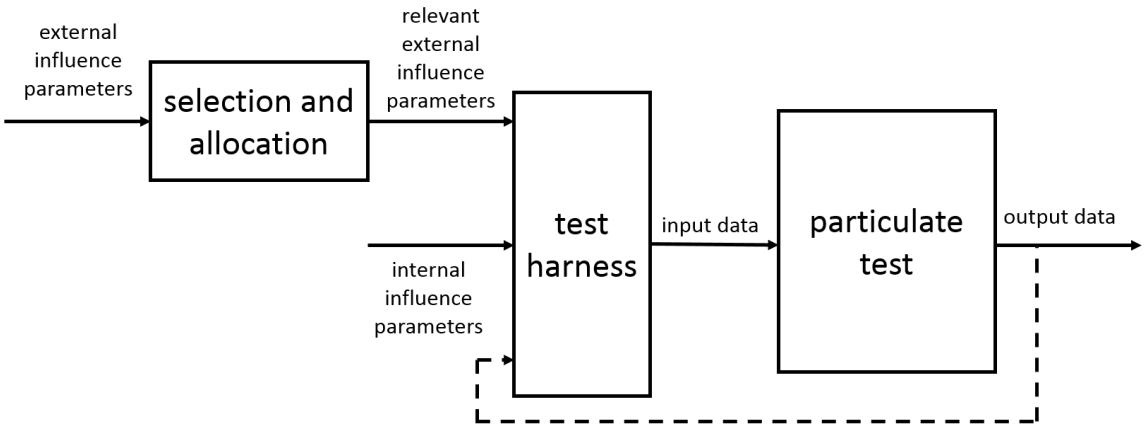


Figure 6-4: Generation of input data for particulate tests.

# 7 Generation of Particulate Test Cases

In this chapter, research question five is addressed:

Q 5: *How can particulate test cases be defined?*

## 7.1 Systematic Test Case Generation in General

According to Schuldt et al.[254], the procedure of an efficient generation of test cases consists of four steps as illustrated in Figure 7-1.

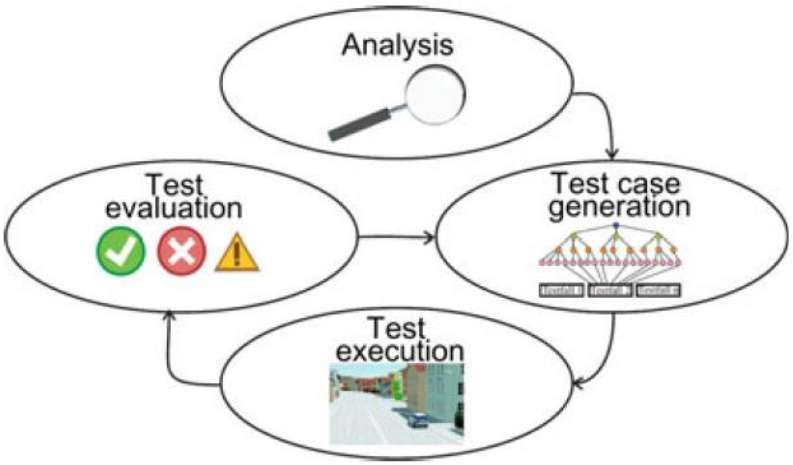


Figure 7-1: Procedure for a test concept according to Schuldt et al.[254, 255]

First of all, the factors that have an influence of the OUT have to be analyzed. The second step is the generic test case generation, which is handled in detail in the remaining part of this chapter. Thereafter the test is executed. Existing methods and environments for test case execution are summarized in subchapter 2.1. In the fourth and final step, the test cases are evaluated. [254]

Schuldt et al. [254] propose a systematic generation of test cases in three steps as illustrated in Figure 7-2.

[254] Schuldt, F. et al.: Effiziente systematische Testgenerierung für FAS in virt. Umgebungen (2013).

[255] Translated illustration: Wachenfeld, W.; Winner, H.: The Release of Autonomous Vehicles (2016), p. 430.

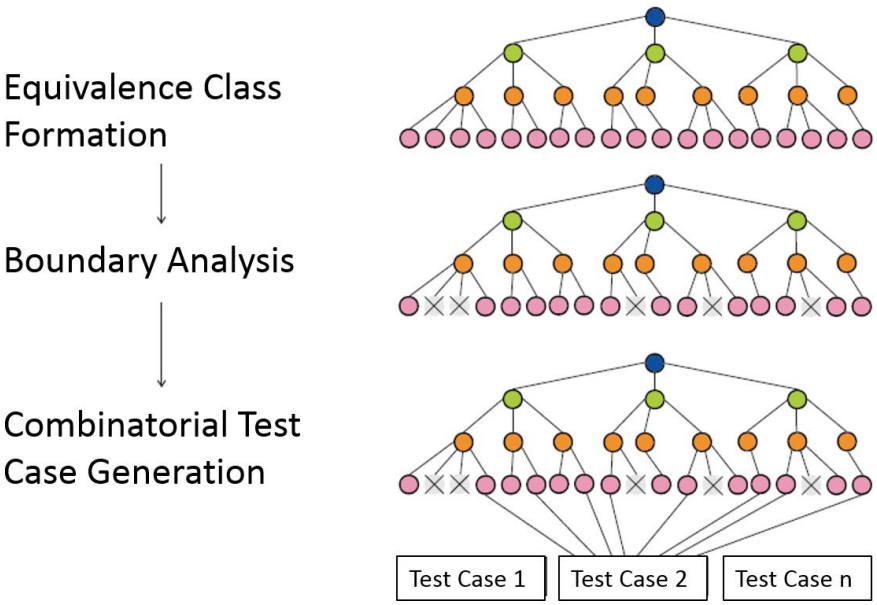


Figure 7-2: Systematic test case generation.[256]

To format equivalence classes in a first step, the value range of the influence factors is split into partitions or equivalence classes. From each of these equivalence classes, an arbitrarily chosen number of parameter values is selected. Hereby the portioning can be partial (i.e. including only valid values) or complete (includes invalid values as well). As the robustness of the OUT can be tested with invalid values, a complete portioning is preferable.[257]

Whereas the values within the equivalency classes for test case generation are randomly chosen, in boundary analysis edge value tuples are chosen. They are a valuable extension of the test suite; however, they increase the number of test cases depending on the number and dimension of equivalency classes.[257]

As *N*-wise coverage, also known as exhaustive testing is not feasible due to the high number of influence factors (cp. section 2.1.4), Schuldt et al.[257] propose to use test suites with *t*-wise coverage. There are different algorithms that are suitable for the generation of *t*-wise test suites described in literature. Grindal et al.[258] as well as Kuhn et al.[259] and Nie and Leung[260]

[256] Translated from Schuldt, F. et al.: Effiziente systematische Testgenerierung für FAS in virt. Umgebungen (2013).

[257] Schuldt, F. et al.: Effiziente systematische Testgenerierung für FAS in virt. Umgebungen (2013).

[258] Grindal, M. et al.: Combination testing strategies: a survey (2005).

[259] Kuhn, D. R. et al.: Practical combinatorial testing (2010).

[260] Nie, C.; Leung, H.: A survey of combinatorial testing (2011).

summarize exiting algorithms for combinatorial testing. Schuldt recommends the algorithm *IPOG* ("*in-parameter-order-general*")[261] for systematic test case generation.

Tao et al.[262] also use the *IPOG* algorithm to create combinatorial test suites for virtual tests of an AEB (autonomous emergency braking) system.

The *IPOG* algorithm starts with a *t*-wise test suite for the first *t* parameters and uses horizontal and vertical growth to extend the test suite until in fulfills *t*-wise coverage for the first *t*+1 parameters in the next set. This is subsequently repeated until the *t*-wise criterion is fulfilled for all *N* parameters. The algorithm is deterministic, thus it always produces the same *t*-wise test suite for a given parameter space.[261]

Schuldt et al.[263] finally propose an automated evaluation of the executed test cases, as due to the high number of test cases a manual evaluation would not be efficient.

## 7.2 Systematic Generation of Particulate Test Cases

Transferring the approach by Schuldt et al.[263] to the generation of particulate test cases where the OUT is one single functional layer of a HAD function, first of all the influence parameters have to be analyzed and allocated to the functional layers. This step is handled in detail in chapter 6.

For systematically generating parameter combinations, the required test coverage has to be defined (compare sections 2.1.4 and 8.2). Thereafter, the same algorithms for generating combinatorial test suites that are used for system tests, e.g. *IPOG* can be applied to generate the test suites for particulate testing.

For the test execution, the most suitable test environment (compare section 2.1.2) has to be chosen for each particulate test case. Hereby, available tools and their maturity and validity have to be considered and compared with the requirements on the test execution. According to Schuldt et al.[263], virtual test environments are always preferable, as they are efficient, reproducible and safely executable. However, for some test cases there are no simulation models e.g. for environment perception sensors available that fulfill all requirements regarding model fidelity.[264] Thus, those test cases either have to be executed in HiL environments with real sensors or as open-loop test by replaying recorded sensor raw data.

---

[261] Lei, Y. et al.: IPOG/IPOG-D: efficient test generation for multi-way combinatorial testing (2008).

[262] Tao, J. et al.: On the Industrial Application of Combinatorial Testing for AD Functions (2019).

[263] Schuldt, F. et al.: Effiziente systematische Testgenerierung für FAS in virt. Umgebungen (2013).

[264] Holder, M. et al.: Challenges in Radar Sensor Modeling for Virtual Validation of AD (2018).

Schuldt et al.[265] propose a 2-step method for the allocation of test cases to XiL methods that is based on the classification of XiL methods with Kiviat diagrams (compare section 2.1.2). This method is further developed by Schuldt.[266] In the first step, available XiL methods are classified in Kiviat diagrams. Thereafter, XiL methods that are suitable for the test case are selected by comparing the Kiviat diagrams of the XiL methods with the requirements for the test case. In the second step, the optimum XiL method for the test case is selected with an evaluation function. Hereby, time, costs and further evaluation criteria, such as quality of simulation models are considered and weighted in a quality function. The method is summarized in Figure 7-3.

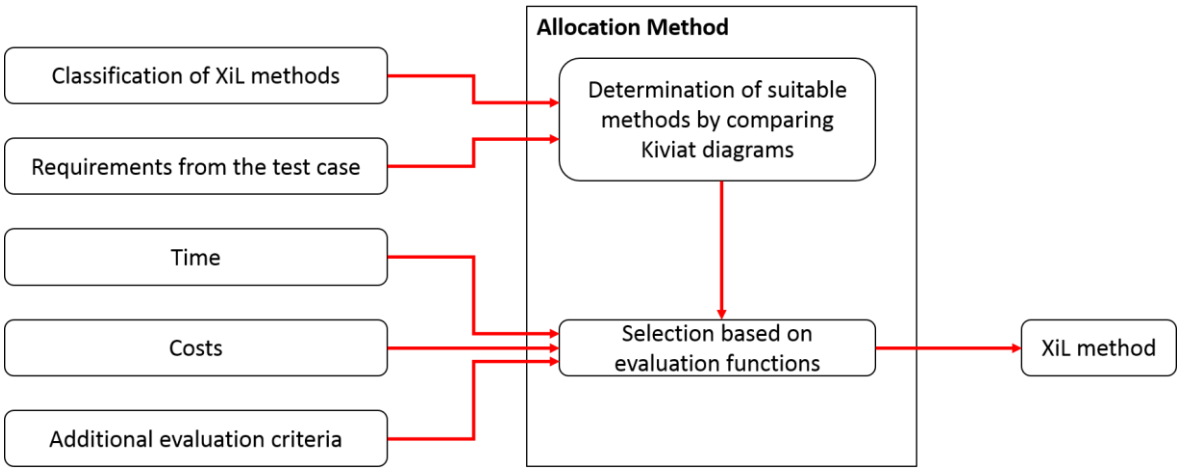


Figure 7-3: Allocation of test cases to XiL methods according to Schuldt.[267]

Böde et al.[268] introduced a method for a quantitative analysis of splitting test cases into real and virtual tests. Thereby, they consider costs as well as model validity and other requirements. This analysis could be used as an additional input for the second step of the allocation method described above.

Finally, the conducted test cases have to be evaluated and documented. The derivation of the therefore required evaluation criteria for particulate testing is handled in detail in chapter 5.

---

[265] Schuldt, F. et al.: Zuordnung von Testfällen auf X-in-the-Loop Verfahren (2015).

[266] Schuldt, F.: Diss., Ein Beitrag für den methodischen Test von autom. Fahrfunktionen (2017), pp. 141–144.

[267] Based on Schuldt, F.: Diss., Ein Beitrag für den methodischen Test von autom. Fahrfunktionen (2017), p. 141.

[268] Böde, E. et al.: Efficient Splitting of Test and Simulation Cases (2018).

## 7.3 Interim Conclusion

There are existing methods for the generation of test cases on system level that can be transferred to generate particulate teste cases. The prerequisites for that are the identification and allocation of parameters with influence on the respective functional layers (see chapter 6) as well as the derivation of particulate evaluation criteria (see chapter 5). Furthermore, the required test coverage has to be defined which is a prerequisite for any kind of validation approach, but not solved yet for scenario-based validation of HAV. Once the required test coverage is known, there are various algorithms existing that can be used for the systematic generation of test suites that fulfill the coverage criterion. Finally, the adequate test environment for each test case can be chosen by comparing the available options and their properties with the requirements of the test case. However, to use the method efficiently in practice, the complete test chain has to be automated which is still an open research need.

# 8 Quantification of the Potential to Reduce the Validation Effort

In this chapter, the last research question is answered:

Q 6: *How high is the potential for reducing the validation effort of HAV by functional decomposition?*

In order to do so, first of all the problem of parameter space explosion is analyzed in detail. Thereafter, the required test coverage for scenario-based validation is estimated as a reference. Finally, the effects that lead to a reduction of test effort with functional decomposition are analyzed and the reduction potential is quantified.

## 8.1 Parameter Space Explosion in Scenario-Based Validation[269]

The scenario-based approach as introduced in section 2.1.1.2 can potentially reduce the approval effort for HAV. However, it still leads to a huge number of concrete scenarios that have to be evaluated in test cases, even for one single logical scenario. For example, in its safety report Waymo[270] states they "*create thousands of variations*" of one single scenario. This so-called parameter space explosion is mainly caused by two contributing factors, namely influence parameters and combinatorics. These are addressed in the following sections:

### 8.1.1 Influence Parameters

Even simple logic scenarios are described by a multitude of parameters that affect the OUT. In section 2.1.1.2.2, a 6-layer model to structure those influence parameters is summarized (cp. Figure 2-2). Schuldt[271] describes how those influence parameters can be selected and analyzed and states different information sources such as regulations for road constructions, vehicle catalogs, or expert knowledge.

[269] This subchapter is based on Amersbach, C.; Winner, H.: Functional Decomposition to Overcome the Parameter Space Explosion (2019).

[270] WAYMO: On the Road to Fully Self-Driving (2017).

[271] Schuldt, F.: Diss., Ein Beitrag für den methodischen Test von autom. Fahrfunktionen (2017), pp. 85–102.

For each parameter $p_i$, $k_i$ different values can be assigned. As the majority of the parameters have a continuous parameter space (e.g. speed), $k_i$ depends on the chosen discretization step width. Choosing the right discretization step width is a challenge yet to be solved, as a too coarse discretization on the one hand will lead to gaps in the parameter space that might lead to undiscovered issues during testing. A too fine discretization on the other hand will lead to a higher number of test cases than necessary. It is not attempted to solve the discretization challenge completely here. However, boundary conditions for the discretization are analyzed in subchapter 8.5. In order to estimate the potential to reduce the validation effort by functional decomposition, values for $k_i$ are assumed. The assumed discretization for some exemplary parameters is summarized in Table 8-1.

Table 8-1: Assumed discretization for exemplary parameters.

| **Representation layer**[272] | **Parameter $p_i$** | **Number of discretization steps $k_i$** |
|---|---|---|
| 1<br>Road Level | width lane 1 | 2 |
| | width lane 2 | 3 |
| 5<br>Environment | sun position | 250 |
| | precipitation (rain, snow, etc.) | 10 |

For the discretization of the lane widths, two-lane Autobahn sections in Germany are considered and it is assumed that those sections are compliant with the relevant guideline for the construction of motorways[273].[274] For the discretization of the sun position in this example, a range of 360° in azimuth and from -10° (e.g. on a hilltop at sunset or sunrise) to 90° in elevation is considered to be able to detect effects that are caused by indirect blinding. Discretizing this range into segments with 12° in azimuth and 12.5° in elevation would lead to 240 instances for the sun position. To allow for finer discretization in the area of direct blinding effects, 250 instances for the sun position are used for the following considerations. For precipitation, 10 different instances are assumed to consider different intensity as well as different raindrop or snowflake sizes.

The complete list of influence parameters - including the values for $k_i$ and the allocation to functional layers - for the exemplary set of scenarios (cp. section 8.3.1) can be found in Appendix A.1.

[272] According to Bagschik, G. et al.: Ontology Based Scene Creation for the Development of Automated Vehicles (2018).

[273] FGSV: Richtlinien für die Anlage von Autobahnen (RAA) (2008).

[274] Cp. Schuldt, F.: Diss., Ein Beitrag für den methodischen Test von autom. Fahrfunktionen (2017), pp. 110–112.

### 8.1.2 Combinatorics

A scenario that is defined by $N$ parameters $(p_1, \dots, p_i, \dots, p_N)$ with $k_i$ instances per parameter will lead to

$$S_N = \prod_{i=1}^{N} k_i \tag{8-1}$$

theoretically possible test cases.[275a] Thus, with each additional parameter or discretization step, $S_N$ progressively increases. Due to time and cost limits, it is not possible to test all theoretically possible test cases.[276] Therefore, a systematic test case generation is required. Grindal et al.[275b] give an overview of combination strategies that can be used for a systematic test case generation. However, as one requirement for test case generation for HAD according to Wachenfeld and Winner is reproducibility[277], only deterministic combination strategies can be used in this domain. The most important constraint for choosing a combination strategy is the required test coverage criterion. For the following considerations, the so-called *t*-wise coverage is used as a coverage criterion as proposed by Schuldt[278]. Grindal et al.[275c] define *t*-wise coverage as follows:

*For t-wise coverage "*[…] *every possible combination of all* […] *values of t parameters* [has to] *be included in some test case in the test suite."*

All possible combinations of all values of all parameters as calculated in (8-1) correlate to *N*-wise coverage, a special case of *t*-wise coverage (cp. section 2.1.4)

Kuhn et al.[279] deduce that the size of a test suite with *t*-wise coverage can be estimated with $k^t$ for the simple case that every parameter $p_i$ has the same number of possible values $k_i = k$ and the test case generation is "perfectly efficient", i.e. there are no duplicates in the test suite. However, for a real application, the assumption that each parameter has the same number of possible values is not valid, as for example the width of a traffic lane has fewer discretization steps than the sun position. Therefore, the derivation above is combined with (8-1) to calculate the size $S_t$ of a test suite with *N* parameters and *t*-wise coverage as following:

$$S_t = \prod_{i=1}^{t} \max_i(k_1, \dots, k_N) \tag{8-2}$$

---

[275] Grindal, M. et al.: Combination testing strategies: a survey (2005).a: p. 169; b: -; c: p. 171 f.

[276] Sommerville, I.: Software engineering (2006), p. 539.

[277] Wachenfeld, W.; Winner, H.: The Release of Autonomous Vehicles (2016), p. 433.

[278] Schuldt, F.: Diss., Ein Beitrag für den methodischen Test von autom. Fahrfunktionen (2017).pp. 124 f.

[279] Kuhn, D. R. et al.: Software fault interactions and implications for software testing (2004).

hereby $\max_i(k_1, \dots, k_N)$ is defined as the $i$-greatest element of the set $(k_1, \dots, k_N)$.

Thus, with a given value $t$ for the required coverage, only the $t$ parameters with the greatest number of discretization steps have an influence on the size of the test suite. Therefore, $S_t$ can be significantly reduced by eliminating the parameters with the finest discretization or by reducing the number of discretization steps (see 8.1.1).

## 8.2 Estimating Required Test Coverage for Scenario-Based Validation[280]

One crucial part of the approval process for HAV is the safety argumentation. In order to reach a general acceptance and confidence, it has to be argued that the performed approval is correct and sufficient to prove that the HAV is safe enough. This argumentation would already be a challenge for *N*-wise testing as the detail level of scenario description (i.e. the number of parameters that describe a scenario) and parameter discretization are more or less arbitrarily chosen (cp. section 2.1.1.2.2 and Table 2-1). Nevertheless, they have a huge influence on the size of the parameter space. As *N*-wise testing is not feasible, the argumentation becomes even more difficult. For *t*-wise coverage, the sufficient value for *t* has to be chosen, which is – in software testing - typically done based on empirical data analysis.[281] However, for HAV there are not enough data available yet to perform empirical analyses. In addition, data of today's traffic are not directly transferable for HAV, as they do not behave similarly to human drivers in all circumstances. For example, it is expected that HAV would drive more defensive than most human drivers would and therefore will stimulate more cutting maneuvers. The same problem emerges when defining threshold values for criticality metrics, which are not necessarily the same for HAV and human drivers. Additionally, the application of a criticality metric includes the simulation of the scenarios to be examined and is therefore as time-consuming as using the scenarios for simulation-based validation. For other test suite generation methods such as randomly distributed test suites (e.g. Monte Carlo experiments), the required test coverage has to be defined and argued to be adequate as well.

As there is no method existing that specifies the required test coverage for scenario-based validation of HAV, the test coverage from the distance-based, statistical approach (cp. section 2.1.1.1) is transferred to the scenario-based approach. Therefore, it is assumed that the parameter distribution in the test suite is equivalent to the parameter distribution in real traffic.

---

[280] This subchapter is based on Amersbach, C.; Winner, H.: Test Coverage for Scenario-Based Validation of HAV (2019).

[281] Kuhn, D. R. et al.: Software fault interactions and implications for software testing (2004).

In order to transfer the reference distance $\bar{s}_{\mathrm{ref}} \approx 7 \cdot 10^8$ km to a reference number of concrete scenarios $n_{\mathrm{ref}}$, the average distance that is covered by one scenario $\bar{s}_{\mathrm{sc}}$ has to be estimated. Therefore, the average timely duration of one scenario $\bar{\tau}_{\mathrm{sc}}$ as well as the average velocity $\bar{v}_{\mathrm{sc}}$ are used:

$$\bar{s}_{\mathrm{sc}} = \bar{\tau}_{\mathrm{sc}} \cdot \bar{v}_{\mathrm{sc}} \tag{8-3}$$

For this example, $\bar{\tau}_{\mathrm{sc}} \approx 7.5$ s and $\bar{v}_{\mathrm{sc}} \approx 30\ \frac{\mathrm{m}}{\mathrm{s}}$ are assumed, based on an analysis of the highD dataset[282], and $\bar{s}_{\mathrm{sc}} \approx 225$ m is derived.

However, as the dividing line between two consecutive scenarios cannot be clearly specified (e.g. the transition between lane change and following is fluent) there is some overlap between two consecutive scenarios. Thus, an overlap factor $f_{\mathrm{o}}$ to calculate the total number of concrete scenarios $n_{\mathrm{tot}}$ that are included in $\bar{s}_{\mathrm{ref}}$ is introduced:

$$n_{\mathrm{tot}} = f_{\mathrm{o}} \frac{\bar{s}_{\mathrm{ref}}}{\bar{s}_{\mathrm{sc}}} \tag{8-4}$$

As it is nearly impossible to estimate the average time or distance fraction with overlapping scenarios, the conservative assumption that there are always two overlapping scenarios, i.e. $f_{\mathrm{o}} = 2$, is made. Furthermore, the fact that one specific concrete scenario might be included multiple times in $\bar{s}_{\mathrm{ref}}$ has to be considered. Therefore, a uniqueness factor $f_{\mathrm{u}}$ is defined:

$$f_{\mathrm{u}} = \frac{n_{\mathrm{u}}}{n_{\mathrm{tot}}} \tag{8-5}$$

Hereby $n_{\mathrm{u}}$ is the number of unique concrete scenarios within $n_{\mathrm{tot}}$. To quantify $f_{\mathrm{u}}$, a clear definition for the uniqueness of scenarios as well as an adequate sample of scenarios would be required. Based on the results by Langner et al.[283], who are analyzing the uniqueness of scenarios with auto encoders, the estimation $f_{\mathrm{u}} = 0.2$ is used. As the distance in the analyzed dataset is quite small compared to $\bar{s}_{\mathrm{ref}}$, this estimation is on the conservative side as $f_{\mathrm{u}}$ is supposed to decrease with increasing distance. With these assumptions, $n_{\mathrm{ref}}$ is determined:

$$n_{\mathrm{ref}} = f_{\mathrm{o}} f_{\mathrm{u}} \frac{\bar{s}_{\mathrm{ref}}}{\bar{s}_{\mathrm{sc}}} \approx 1.2 \cdot 10^9 \tag{8-6}$$

According to Wachenfeld and Winner[284], $\bar{s}_{\mathrm{ref}}$ has to be tested with a distance factor $f_{\mathrm{s}}(e, f_{\mathrm{SP}}, P_{\mathrm{suc}})$, which depends on the desired significance level $e$, the estimated safety performance factor $f_{\mathrm{SP}}$ and the desired probability of success $P_{\mathrm{suc}}$. Hereby, $f_{\mathrm{SP}}$ describes the safety performance of the OUT compared to its reference and $P_{\mathrm{suc}}$ is the probability that the

[282] Krajewski, R. et al.: The highD Dataset (2018).

[283] Langner, J. et al.: Estimating the Uniqueness of Test Scenarios using Autoencoders (2018).

[284] Wachenfeld, W.; Winner, H.: The Release of Autonomous Vehicles (2016), pp. 439–442.

test shows that the OUT is actually safer than this reference. Comparable to Wachenfeld and Winner[285], $f_s(e = 5\%, f_{SP} = 2, P_{suc} = 50\%) = 10$ is used here. Transferring this to the scenario-based approach while considering the conservative estimations of $f_o$ and $f_u$, the upper limit for the number of concrete scenarios that are required for the validation is:

$$n_{req} = f_s n_{ref} \approx 1.2 \cdot 10^{10} \tag{8-7}$$

Having determined $n_{req}$, it can be directly used as a target value for the generation of randomly distributed test suites. If *t*-wise test coverage is preferred, $n_{req}$ can be used to select the required value for *t* so that

$$S_t \geq n_{req} \tag{8-8}$$

is fulfilled. The size of the *t*-wise test suite $S_t$, which includes all relevant scenarios, hereby not only depends on *t* but also on the discretization of the parameter space. Therefore, *t* cannot be defined independently from the parameter discretization. Thus, the compromise between fine discretization and a high value of *t* has to be solved. A coarser discretization will result in a higher value for *t* and thus failures with a higher FTFI (failure triggering fault interaction) number can be unveiled.[286] However, a too coarse discretization might lead to undiscovered failures in the parameter space between the sampling points. The correlation between t and the level of discretization is analyzed in more detail in subchapter 8.5.

## 8.3 Reduction Potential with Functional Decomposition[287]

Amongst other potential benefits that are outlined in section 3.1.3, particulate testing is expected to shrink the relevant parameter space significantly and therefore to reduce the number of required test cases. This is mainly based on these three effects:

1. The parameter space for one single functional layer is smaller than the parameter space for the complete HAD function.
2. Less complex subsystems require a smaller test coverage.
3. The test of perception layers can be aggregated for a set of scenarios.

[285] Wachenfeld, W.; Winner, H.: The Release of Autonomous Vehicles (2016), pp. 441–442.

[286] Kuhn, D. R. et al.: Software fault interactions and implications for software testing (2004).

[287] This subchapter is based on Amersbach, C.; Winner, H.: Functional Decomposition to Overcome the Parameter Space Explosion (2019).

In the following sections, an exemplary set of logical scenarios is used to analyze those effects and to quantify the potential to reduce the validation effort for HAV by functional decomposition and particulate testing.

### 8.3.1 Exemplary Scenario Set

For the analysis, a set of nine logical scenarios for the validation of a highway chauffeur[288] (SAE Level 3[289]) is used. This scenario set is illustrated in Figure 8-1 and the corresponding parameter space and the assignment of the parameters to the functional layers is documented in Appendix A.1.

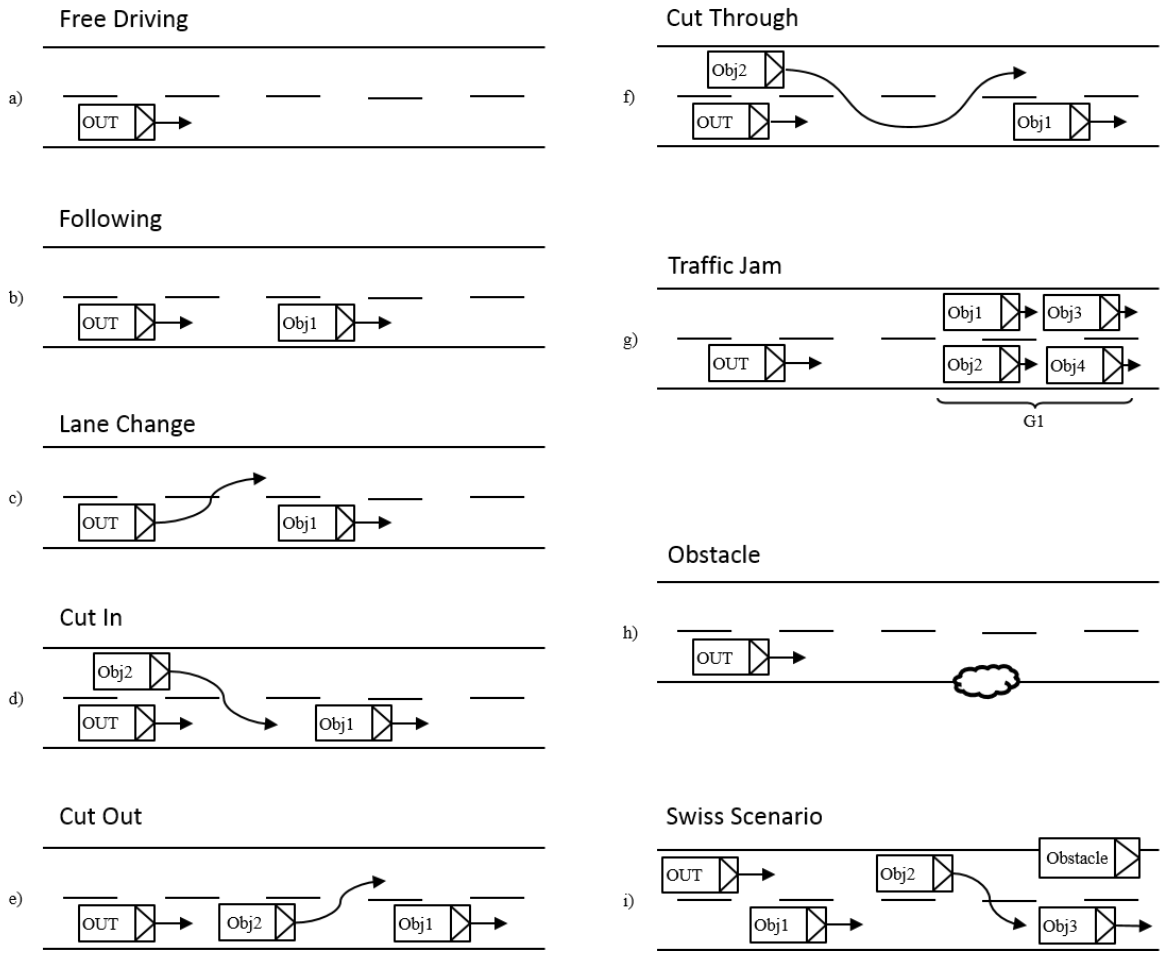


Figure 8-1: Exemplary scenario set.

While the "Swiss scenario" is a representation of a scenario that led to a real-world accident of a Tesla Model S in Switzerland in May 2016[290], the other scenarios are used as baseline

[288] PEGASUS Project Office: PEGASUS Abschlussveranstaltung 2019 - The Highway-Chauffeur (2019).

[289] SAE: J3016: Taxonomy and Definitions for […] Vehicle Automated Driving Systems (2018).

[290] Lambert, F.: Tesla Model S driver crashes into a van while on Autopilot [Video] (2016).

scenarios within the project PEGASUS. The choice of the parameters and their allocated discretization steps is slightly arbitrary as it is based on assumptions and expert knowledge due to the lack of empirical data. However, choosing different parameters and discretization steps would change the absolute values, but not affect the relative comparison between particulate testing and testing of the complete system. The exemplary scenario set in total would lead to $S_N \approx 10^{31}$ possible test cases according to equation (8-1). Figure 8-2 shows the sizes of test suites with different *t*-wise coverage for some of the exemplary scenarios according to equation (8-2). For small values of *t*, the complexity of the scenario has no notable influence on the size of the test suite. Even for high test coverages, the size of the test suites for scenarios with different complexity is within one order of magnitude.

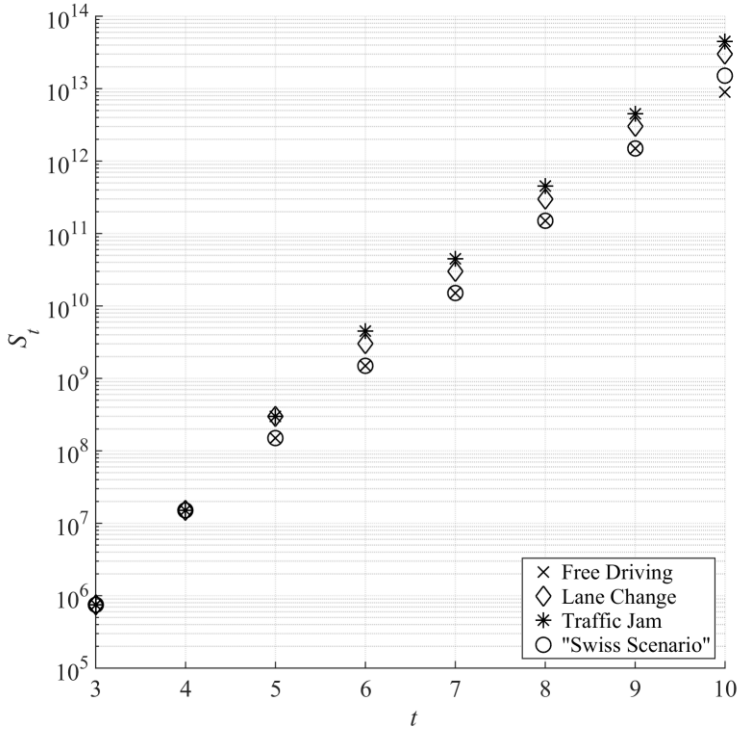


Figure 8-2: Size of *t*-wise test suites for exemplary scenarios.

Applying equation (8-8) and the assumptions made in subchapter 8.2 would require at least a 6-wise coverage for the exemplary scenario set. However, as the relation between $S_t$ and $t$ strongly depends on the detail level of scenario description and the discretization of influence parameters (cp. subchapter 8.2), for the following analysis the range from 3-wise to 10-wise coverage is considered.

### 8.3.2 Particulate Parameter Spaces

Most of the influence parameters only have an influence on some of the layers, e.g. the majority of the environment representation parameters only affect the functional layers 1 and 2. Therefore, the particulate parameter spaces, i.e. the influence parameters for each functional layer are subsets of the total parameter space. However, this only has an effect if parameters

with high numbers of discretization steps are not relevant for some of the layers (compare section 8.1.2). The allocation of influence parameters to functional layers is done based on expert knowledge (compare chapter 6) for this analysis. This effect leads to a reduction of the size of the test suite for higher test coverages (i.e. $t \geq 6$) of around 40 - 50 % for the exemplary scenarios analyzed in this study. However, for lower test coverages (i.e. $t \leq 2$), the size of the test suite will be increased.[291] Nevertheless, as a particular test most likely requires less effort compared to a system test, the total test effort can still be reduced. Figure 8-3 summarizes the effect of reduced particulate parameter spaces on the test suite size for a range of *t*-wise test coverages.

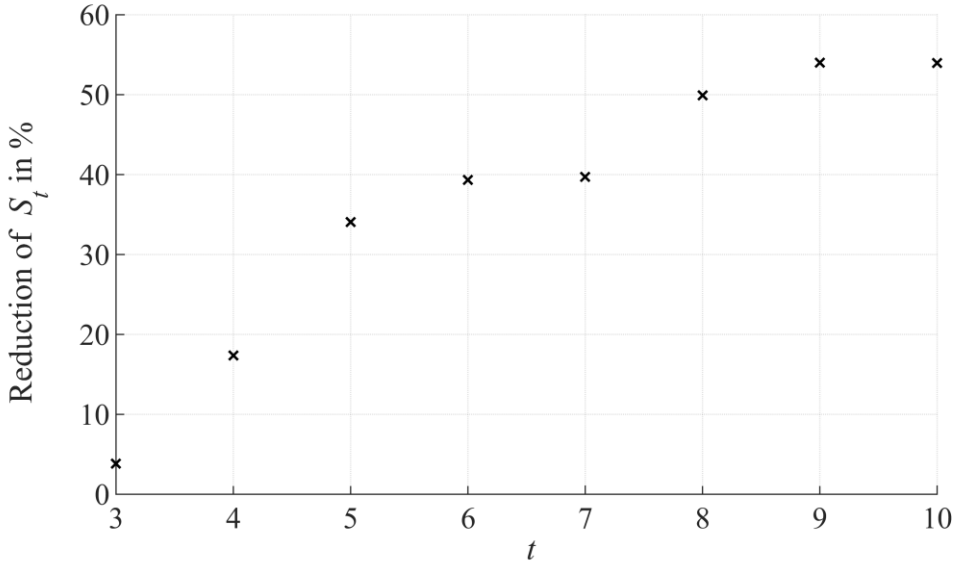


Figure 8-3: Reduction of the test suite size by parameter space reduction through functional decomposition for different *t*-wise test coverages.

### 8.3.3 Less Complex Subsystems Require a Smaller Test Coverage.

As the maximum FTFI number of all failures within a system depends on the complexity of the system, it is expected that the maximum FTFI number of a subsystem is lower than the maximum FTFI number of the complete system and therefore the required coverage for particulate testing is smaller compared to testing of the complete system. However, the required test coverage for a new system cannot be predicted and empirical analysis of the FTFI numbers for HAD functions and their subfunctions is not possible yet, as not enough data are available. This effect leads to a reduction of the parameter space by one order of magnitude

[291] For particulate testing, one system test case is split into one particulate test case for each functional layer, i.e. six particulate test case in total. Thus, without considering effects that lead to a reduction of test cases, the size of the test suite is increased by a factor of six. For low test coverages, this is dominant over the parameter space reduction and thus the number of test cases is increasing.

if the required coverage for particular testing is reduced by a minimum of one (i.e. $t_{\text{part}} \leq t_{\text{sys}} - 1$) compared to testing of the complete system. For the exemplary set of scenarios, this means reducing the required test coverage from 6-wise for system testing to 5-wise for particulate testing. This reduces the absolute test coverage, but not the relative coverage, which is even increased for some functional layers. A 6-wise coverage for the Swiss scenario (Figure 8-1 i)) with 33 parameters incorporates the interaction amongst 18 % of the influence parameters. Whereas a 5-wise coverage of the functional layers in the same scenario corporates the interaction amongst 18% (for layer 1 with 28 influence parameters) to 83% (for layer 5 with 6 influence parameters) of the influence parameters.

### 8.3.4 Aggregation of Perception Layer Tests

When analyzing the influence parameters, it becomes evident that the majority of the parameters with a high number of possible values only have an influence on the perception of the HAD function that is represented in the functional layers 0-2.

An obvious example is the sun position, which only influences layers one and two. Additionally, its possible parameter space is rather big, considering that the azimuth angle (relative to the OUT) can obtain values between 0° and 360° while the elevation angle is spread between around -10° and 90°, depending on the topology and location. Here one could argue that only sun positions within the field of view of cameras and LIDAR have to be considered, which might be true when only evaluating direct blinding effects. However, to include perception errors caused by sun reflections from static or dynamic objects in the scene, all possible sun positions in combination with object attributes and positions have to be included in the test suite.

Aggregating the parameters with influence on the perception layers in one equivalency class scenario for a set of similar scenarios (e.g. scenarios on a two-lane Autobahn) that contains the parameter space for the complete scenario set could thus further reduce the size of the test suite. The affected layers only have to be tested in the equivalency class scenario, which contains all possible combinations of all parameters with influence on those layers. In the exemplary scenario set analyzed in this study, the size of the test suite was reduced by over 50% for small test coverages by this approach. However, for 9-wise or higher coverage, the size of the test suite was increased (see Figure 8-4). This can be explained by the fact that the equivalency class scenario has a higher complexity than the scenarios from the set and therefore contains more influence parameters with a high number of possible values.

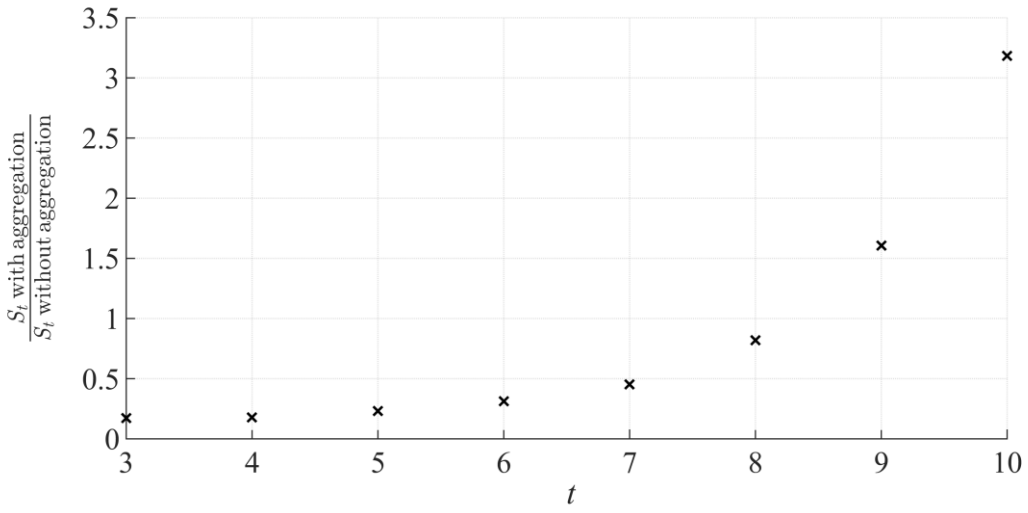


Figure 8-4: Reduction of the test suite size by aggregation of parameters in an equivalency class scenario.

### 8.3.5 Total Reduction of the Test Suite Size by Functional Decomposition

The combination of all effects discussed above leads to a reduction of the entire test suite - consisting of the nine exemplary scenarios (cp. section 8.3.1) by a factor between around 130 for 3-wise coverage and around 20 for 10-wise coverage as illustrated in Figure 8-5. This means that the size of the required test suite was reduced by 99% - 95% by particular testing compared to scenario-based testing of the complete system. For the analyzed set of scenarios and the estimated required test coverage from subchapter 8.2, the size of the required test suite can be reduced by around two orders of magnitude to approximately $1.2 \cdot 10^8$ test cases.

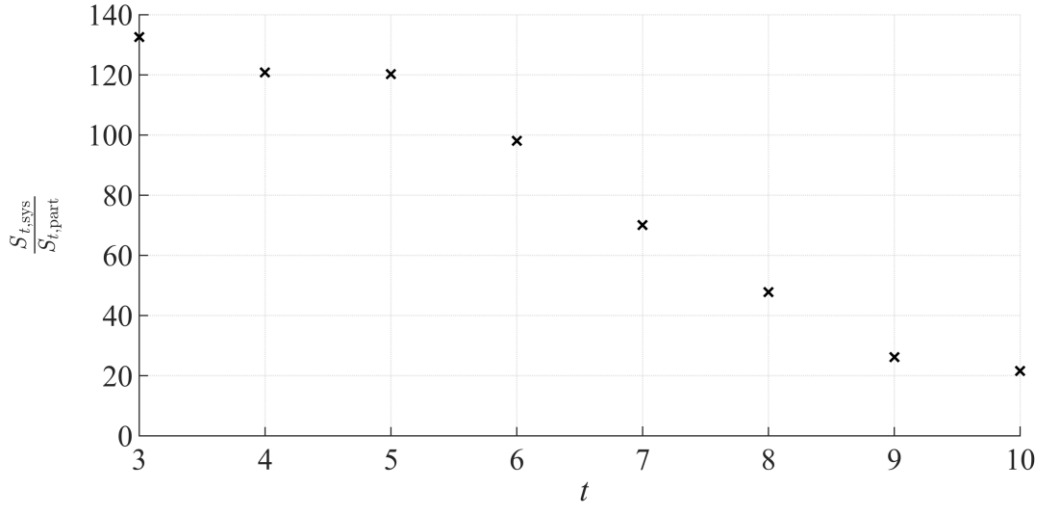


Figure 8-5: Potential reduction of the test suite size by functional decomposition.

## 8.4 Comparing the Reduction Potential with Feasible Test Effort[292]

To estimate whether the required test coverage that remains with functional decomposition is feasible, some additional assumptions need to be made. As in subchapter 8.2, the average duration of one scenario is estimated to $\bar{\tau}_{\mathrm{sc}} \approx 7.5$ s. It is assumed that all tests can be carried out in simulation. Thus, feasible values for the real-time factor $f_{\mathrm{rt}}$ and the parallelization factor $f_{\mathrm{p}}$ have to be assumed. The real-time factor strongly depends on the detail level of the simulation and the computational hardware used. Hereby, a realistic simulation of the environment perception with active sensors, such as radar and lidar, is the most computationally intensive and leads to low values for $f_{\mathrm{rt}}$ around one in the best case if high-end computational hardware is used.[293] With simplified or ideal sensor models, higher values for $f_{\mathrm{rt}}$ can be achieved at the expense of reduced fidelity and validity. Those models are therefore not sufficient for all validation tests.[294] Thus, $f_{\mathrm{rt}} = 1$ is assumed. The attainable parallelization factor is only limited by the used computational hardware and could - in theory - be increased to infinity. However, this is not feasible in practice. It is assumed that $f_{\mathrm{p}} = 1000$ is a feasible approximation as it lies in the same order of magnitude as the number of prototypes that are used for the validation of current vehicles.[295] With these assumptions, the total time $\tau_{\mathrm{tot}}$ that has to be spent for the validation of one version of the OUT can be estimated:

$$\tau_{\mathrm{tot}} = \frac{S_t \bar{\tau}_{\mathrm{sc}}}{f_{\mathrm{rt}} f_{\mathrm{p}}} \approx 1.5 \text{ weeks} \tag{8-9}$$

However, it has to be considered that all tests have to be repeated if an error is detected and fixed and therefore a new version of the OUT is created and thus multiple iterations have to be tested. Additionally, this estimation includes that all tests are carried out in virtual environments and that the test execution as well as the evaluation of the test cases are done fully automatedly and thus the simulations can run 24/7. Nevertheless, even a 10 times longer total simulation time of 15 weeks for one iteration still seems to be feasible.

---

[292] This subchapter is based on Amersbach, C.; Winner, H.: Test Coverage for Scenario-Based Validation of HAV (2019).

[293] Holder, M. et al.: Challenges in Radar Sensor Modeling for Virtual Validation of AD (2018).

[294] Cao, P. et al.: Perception sensor modeling for virtual validation of automated driving (2015).

[295] Christiansen, M.: In geheimer Mission: auf Abnahmefahrt mit der neuen Mercedes E-Klasse, W213 (2015).

# 8.5 Boundary Conditions for Discretization of Influence Parameters

In this subchapter, the correlation between discretization and required *t*-wise coverage is analyzed in more detail and boundary conditions for the discretization of influence parameters are derived.

For a given number of required test cases $n_{\mathrm{req}}$, the required value for t is derived by solving the inequation (8-8): $S_t \geq n_{\mathrm{req}}$ (cp. subchapter 8.2). Hereby, $S_t$ is depends on the discretization of the influence parameters, expressed by the number of instances per parameter $k_i$ (cp. equation (8-2)). To analyze this effect, the discretization of the exemplary parameter space (cp. section 8.3.1 and appendix A.1) is altered. Therefore, $k_i$ is multiplied with a discretization factor $f_{\mathrm{d}}$. For this analysis, the same discretization factor $f_{\mathrm{d}}$ is applied to all parameters. Thus, equation (8-2) is extended as following to determine the size of a *t*-wise test suite with modified discretization $S_{t,d}$:

$$S_{t,d} = \prod_{i=1}^{t} f_{\mathrm{d}} \cdot \max_i(k_1, \dots, k_N) = f_{\mathrm{d}}^t \prod_{i=1}^{t} \max_i(k_1, \dots, k_N) \tag{8-10}$$

Using the size of the test suite with the original discretization as reference $S_{t,1}$, the influence of $f_{\mathrm{d}}$ becomes clearer:

$$S_{t,\mathrm{d}} = f_{\mathrm{d}}^t \cdot S_{t,1} \tag{8-11}$$

Using equation (8-11), inequation (8-8) is solved for different discretization factors $f_{\mathrm{d}}$ to determine the related values for *t*.

However, as *t* can only obtain integer values, $S_{t,\mathrm{d}}$ has discrete steps. Thus, for some combinations of *t* and $f_{\mathrm{d}}$, $S_{t,\mathrm{d}}$ is multiple times greater than $n_{\mathrm{req}}$. To consider this fact, an effort factor $f_{\mathrm{eff}}$ is introduced:

$$f_{\mathrm{eff}} = \frac{S_{t,\mathrm{d}}}{n_{\mathrm{req}}} \tag{8-12}$$

Figure 8-6 depicts the correlation between $f_{\mathrm{d}}$, *t* and $f_{\mathrm{eff}}$ for system tests (index $_{\mathrm{sys}}$) as well as for particulate testing (index $_{\mathrm{part}}$) based on the exemplary scenario set, parameter space and assumptions used throughout chapter 8:

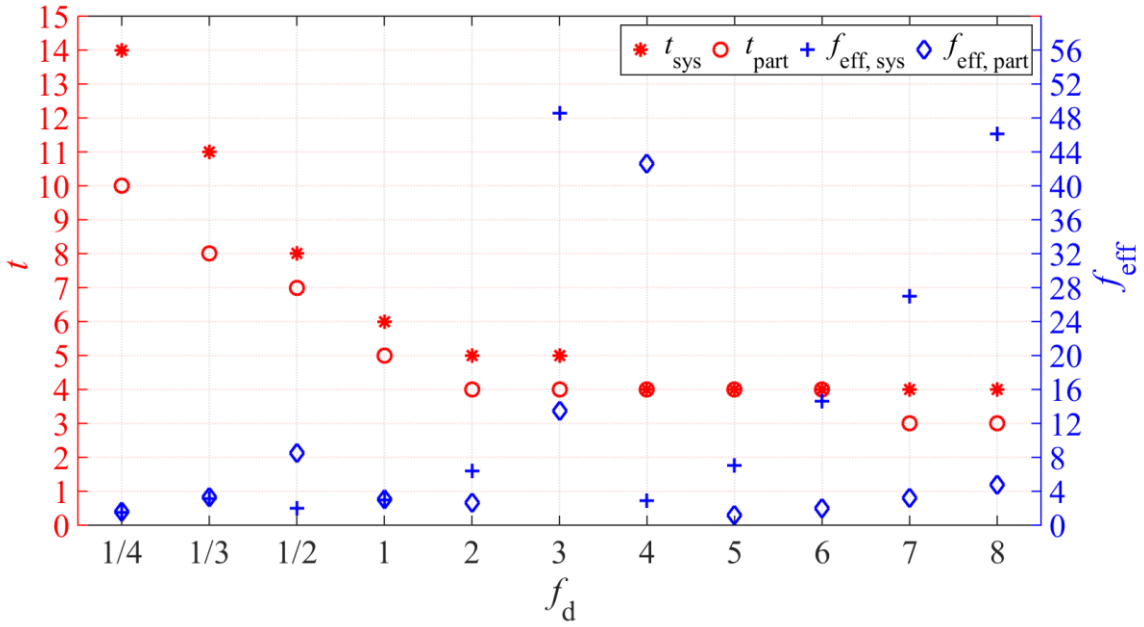


Figure 8-6: Correlation between $f_d$, $t$ and $f_{eff}$.

It can be seen, that for a coarser discretization (i.e. $f_d < 1$) $t$ is increasing as expected for a given number of minimum required test cases $n_{req}$. For a finer discretization, $t$ is decreasing. However, as $S_{t,d}$ has discrete steps, in some areas a finer discretization does not reduce $t$. Instead, $f_{eff}$ is increasing with $S_{t,d}$. For some values of $f_d$, the effort to reach $n_{req}$ is increased by a factor of almost 50. This reduces the reduction potential of functional decomposition significantly. Thus, $f_{eff}$ has to be considered when choosing a discretization. Moreover, altering the discretization can be used to minimize $f_{eff}$ if other boundary conditions allow it.

Besides the correlation between $f_d$, $t$ and $f_{eff}$, the fact that functional decomposition is most effective for values of $t$ below 6 (cp. Figure 8-5) also has to be considered when discretizing parameters for particulate testing.

According to equation (8-2), only the $t$ values with the highest number of instances $k_i$ have an influence on the size of $S_t$. Thus, it is recommended to sort the parameters in descending order of $k_i$ (i.e. $k_{i+1} \leq k_i$). Then in a first step, the discretization of the parameters $p_1$ to $p_t$ can be optimized to minimize $f_{eff}$ and set $t$ to a value below 6. In a second step, the discretization of the remaining parameters $p_{t+1}$ to $p_n$ can be refined so that $k_i = k_t$ (for $i \in [t+1; n]$) to reduce gaps in the parameter space without increasing $S_t$. However, as not all parameters have an influence on all functional layers, this step has to be done after the allocation of parameters for each particulate parameter space rather than for the total parameter space.

## 8.6 Interim Conclusion

In this chapter, the required size of test suites for scenario-based testing and particulate testing for HAV are analyzed and compared for an exemplary set of scenarios. The discretization steps of the influence parameters and the required test coverage mainly influence the size of the test suites. This leads to a "discretization challenge" that is not solved completely within this work. However, the interdependencies between *t*-wise coverage and discretization step width are analyzed and boundary conditions for discretization are derived. Additionally, assumptions are made to create an exemplary parameter space as a basis to quantify the potential to reduce test effort by functional decomposition.

Furthermore, a method to define the required test coverage for the scenario-based validation of HAV has been proposed and exemplarily applied. Using the average distance between fatal accidents in the ODD as a safety reference, an existing statistical approach for distance-based validation is transferred to scenario-based validation. With this method, the number of scenarios that are required to create equal evidence as the distance-based approach can be specified.

Based on those assumptions, the potential to shrink the parameter space by functional decomposition and particular testing of HAD functions is analyzed. While the absolute values for the test suite size strongly depend on the assumptions made throughout their creation, the relative findings are not affected as the same assumed data are used to compare both methods, particulate testing and testing of the complete system. However, the findings have yet to be validated with a practical implementation of the functional decomposition approach. Furthermore, it has not been proven yet, to which extent the results from the exemplary scenario set with only nine logical scenarios are transferable to a more extensive scenario catalog.

It is concluded that the functional decomposition approach potentially reduces the size of the required test suite by a factor of 20…130, depending on the required test coverage and parameter discretization.

Additionally, the functional decomposition approach can be combined with scenario selection and reduction methods (cp. section 2.1.5.2). Thus, the test effort can be further reduced, or the reduced scenario catalog can be tested with finer discretization or higher *t*-wise coverage.

# 9 Summary and Conclusion

In this final chapter, a conclusion on the performed research is drawn. Therefore, in subchapter 9.1, the results in the form of answers to the research questions from subchapter 3.2 are summarized and discussed. Thereafter, in subchapter 9.2 they are compared to the requirements and the scope that have been defined in subchapter 3.1.

This is followed by an analysis of the applicability of the method in the automotive industry in subchapter 9.3. Finally, in subchapter 9.4, the remaining research questions are outlined.

## 9.1 Summary of the Results

In chapter 1, current test concepts for ADAS have been introduced. They cannot be transferred without adaption to HAD due to the increased functional scope and the increased safety requirements when getting the human driver completely out of the loop. This motivates the development of new approaches for the validation of HAD. Based on this motivation, the following research theses have been developed:

T 0: *The validation effort for HAV can be reduced by functional decomposition and particulate testing.*

T 1: *A decomposition of highly automated driving functions into functional layers, which are independent of a concrete system architecture, is possible*.

T 2: *It is possible to define particulate test cases and related evaluation criteria that test each functional layer independently from the remaining system and evaluate it on its interfaces.*

Thereafter, in chapter 2, the relevant state of the art has been summarized. There are different approaches for verification, validation and testing of HAV described in literature. All of them, except for formal verification, have in common that they focus on black box system tests that are evaluated based on the observable behavior and that the high required test scopes lead to an unfeasible validation effort. The different test approaches can be combined with various test environments. Thereby, a scenario-based validation in virtual environments seems to be the most promising approach to overcome the *approval trap*. However, due to the high number of influence parameters, the scenario-based approach leads to a parameter space explosion. Existing approaches to reduce the approval effort therefore focus on an efficient selection or generation of relevant scenarios.

Besides verification and validation approaches for HAV, there are also various decomposition approaches for human or automated driving tasks described in literature. The majority of them originates from accident analysis or system architectures.

Based on the research theses and the state of the art, the research objective was concretized in chapter 3 and research questions were derived. Those research questions were answered in chapters 4 to 7. The research questions and answers are summarized in the following sections.

Q 1: *Which independent and generic functional layers are suitable for a functional decomposition of HAD functions?*

A 6-layer functional decomposition scheme that is based on existing decompositions of the human driving task in accident analysis is proposed for the decomposition of HAD functions:

(0) Information Access
(1) Information Reception
(2) Information Processing
(3) Situational Understanding
(4) Behavioral Decision
(5) Action

This decomposition scheme can be mapped to existing reference system architectures of HAD functions.

Q 2: *How can the interfaces between the functional layers derived in Q1 be defined generically?*

The interfaces between the functional layers are based on existing terminology for scenario-based testing and on standardized data formats that are state of the art in virtual testing. However, for some of the interfaces, suitable standards are still in development or have to be developed in the future. Not all of the specified interfaces are currently available for testing in real environments, as they are physically not available yet (e.g. sensor raw data interface between layer 1 and 2). In this case, the affected layers have to be tested and evaluated in combination.

Q 3: *How can particulate evaluation criteria for decomposed HAD functions be derived?*

Particulate evaluation criteria can be derived from safety goals and evaluation criteria on system level. Therefore, existing tools from safety analysis such as FTA or STPA are used. Evaluation criteria have to be formulated in a computable way to allow for an automated test evaluation. Ontologies, skill graphs and rule engines can serve as a basis for an automated derivation of particulate evaluation criteria that has to be developed for a large-scale application of the method.

Q 4: *How can influence parameters be allocated to the functional layers derived in Q1?*

External input parameters have to be selected from different information sources and discretized as a prerequisite. Thereafter, the relevant external influence parameters are allocated to the functional layers. For prototypical implementations, this can be done intuitively, based on expert knowledge. However, for an application in the development and validation pro-

cess, the allocation has to be objectified and automated. To generate input data for the particulate tests from the allocated external influence parameters, appropriate test harnesses are required. Those also have to consider internal influence parameters as well as a feedback from the output data of the functional layer under test.

Q 5: *How can particulate test cases be defined?*

To define particulate test cases, the influence parameters can be systematically combined to test suites by existing combination strategies if the required test coverage has been defined. Thereafter, the adequate test environment has to be selected for each test case. Therefore, an automated derivation of requirements on the test environment is required. For an application of the functional decomposition method in development and validation, furthermore the complete test chain has to be automated.

Q 6: *How high is the potential for reducing the validation effort of HAV by functional decomposition?*

To quantify the potential to reduce the validation effort of HAV by functional decomposition, first, the required test coverage for scenario-based validation is estimated. Additionally, the effects that lead to the so-called parameter-space explosion in scenario-based testing are analyzed. Furthermore, the effects that lead to a reduction of the required test suite sizes by functional decomposition are analyzed and quantified with an exemplary set of scenarios. Depending on the required *t*-wise test coverage and on the discretization of the influence parameters, the size of the test suite can be reduced by a factor between around 130 for low values of *t* and around 20 for high values for *t*. For the exemplarily analyzed scenario set and the estimated required test coverage can be reduced from around $1.2 \cdot 10^{10}$ to $1.2 \cdot 10^{8}$ required test cases by functional decomposition. Under optimal conditions and if all tests could be carried out in an automated virtual test environment, the remaining test suite would require around 1.5 weeks of simulation time.

## 9.2 Meeting the Objective and Requirements

In this subchapter, it is reflected whether the objective and the requirements that have been outlined in chapter 3 are met.

The objective of this dissertation has been to develop a method for the application of functional decomposition for the validation of HAV. Hereby, the focus was laid on the definition of functional layers and interfaces as well as the derivation of particulate evaluation criteria and particulate parameter spaces. The above-mentioned parts of the methodology that are within the defined scope of this dissertation have been developed in general. However, for an application of the method, a more detailed development is required, especially for automating the test chain. Furthermore, not all essential prerequisites for the application of the method are available yet. Besides valid simulation models for environment perception sensors that are required for virtual testing, especially the definition of macroscopic and microscopic safety requirements and the

discretization of the parameter space as well as the definition of the required test coverage are still unsolved research questions.

In the following sections, it is discussed whether the requirements are met:

**Representative-valid**

This requirement depends on the selection and allocation of external influence parameters as well as the selection of test environments for the particulate tests. While the selection of external influence parameters is not within the scope of this work, the transformation of external influence parameters to input data for the particulate test is crucial for meeting the requirement and therefore has to be considered for the selection of the test environments. A key therefore is the automated derivation of requirements on the test environment including test harnesses. This is still an open research question.

Furthermore, the allocation of external influence parameters to functional layers is crucial. If an influence parameter that is relevant for a specific functional layer is not allocated to it, it is also not considered for the generation of particulate test cases. This will lead to gaps in the test suite. The same applies if a relevant scenario is not included in the scenario catalog, which is a generic problem for scenario-based validation.

**Economical**

As it is intended to carry out as many tests as possible in virtual environments and to test functional layers independently from each other, in general the method fulfills this requirement. However, as long as the prerequisites and required tools for an automated application of the method are not available, it has to be analyzed whether the initial effort can be compensated by the benefits.

**Reproducible**

As long as deterministic combination strategies for the test case generation are used and the derivation of evaluation criteria as well as the allocation of influence parameters are done rule-based rather than intuitive by experts, the method is reproducible.

**In good time**

As the functional layers can be tested independently from each other in virtual environments, the method can be applied early in the development phase. Thus, the requirement is met.

**Independent and generic decomposition layers**

The decomposition scheme can be applied to different system architectures. However, there are also system architectures (e.g. end-to-end learning) to which the method cannot be applied or only parts of the method can be applied. Thus, the requirement of generic decomposition layers is not fully met. Whether the decomposition layers can be tested independently depends on the availability of suitable test harnesses and physical interfaces of the hardware components as well as observable interfaces in the software.

**Generic and observable interfaces**

This requirement could not be completely met. Where it is possible, already existing standards are used for the interfaces. Thus, those interfaces are generic and observable if the OUT follows the standard. However, for some of the interfaces it is not possible to define common standards, as there are fundamentally different options for the representation of those interfaces (for example the subjective scene representation). Furthermore, with state-of-the-art hardware, some of the interfaces are not observable, as they do not physically exist.

**Explicit pass/fail criteria for all particulate tests**

In this work, a method to derive particulate evaluation criteria has been outlined. However, as this method has not been applied yet, no statement can be made as to whether it is possible to derive explicit and unambiguous evaluation criteria for all particulate tests.

## 9.3 Applicability in the Automotive Industry

To be able to apply the functional decomposition method in the automotive industry, general prerequisites that are required for the validation of HAV have to be available:

- Macroscopic and microscopic safety requirements on system level have to be defined.
- The required test coverage has to be defined on system level.
- A complete catalog or database with relevant scenarios has to be created.

Furthermore, adequate simulation models and environments for virtual validation have to be available and validated.

A specific requirement for an industrial application of the functional decomposition approach is the availability of test harnesses and interfaces. Once standards for all of the required interfaces are defined, they can be implemented for newly developed HAD functions. Additionally, all steps of the methodology have to be automated or semi-automated, as the manual derivation of test cases is not feasible beyond prototypical applications.

If those requirements are fulfilled, the method itself has to be validated. Thereafter, it can be implemented in development and validation processes. Furthermore, it can be combined with additional methods (cp. section 2.1.5) for validation effort reduction.

## 9.4 Remaining Research Questions

To conclude this dissertation, remaining research questions that have been raised during the execution of this work are summarized:

The questions of discretization of influence parameters as well as of defining the required test coverage for scenario-based testing have already been raised by Schuldt[296]. However, they have not been answered completely yet, despite the fact that they are fundamental for scenario-based validation. While boundary conditions for solving the trade-off between discretization and *t*-wise coverage have been analyzed in this work, the FTFI numbers for HAV are still unknown. Furthermore, a comprehensively accepted criterion to define the required test coverage for scenario-based testing has to be found. A first approach therefore was proposed in this work. However, the definition of the total size of a test suite is not sufficient without specifying a related scenario catalog including scenario and parameter distributions. Furthermore, the emergence of numerical errors, which only occur when exact double values are taken by the influence parameters, has to be analyzed.

Another fundamental question for the validation of HAV is the definition of macroscopic and microscopic safety requirements, i.e. a holistic answer to the question *"How safe is safe enough?"*. A foundation to answer this question is laid by Junietz[297].

A specific research need for the further development of the functional decomposition approach is the automation of all single steps of the methodology. For the automated derivation of particulate evaluation criteria as well as for the automated allocation of influence parameters, the automatic derivation of dependency chains introduced by Hoßbach[298] could be used as a basis.

Another question that has to be solved as a basis for a selection of suitable test environments is the derivation of requirements regarding fidelity on test environments, simulation models and test harnesses for individual particulate tests.

Finally, a holistic proof of applicability and validity of the functional decomposition method must be provided. Thereby, the quantification of the potential to reduce approval effort for HAV by functional decomposition can be validated as well.

---

[296] Schuldt, F.: Diss., Ein Beitrag für den methodischen Test von autom. Fahrfunktionen (2017), pp. 85–102.

[297] Junietz, P.: Diss., Microscopic and Macroscopic Risk Metrics (Announced for 2019).

[298] Hoßbach, P. M.: Masterthesis, Automatic Derivation of Dependency Chains (2019).

# A Appendix

## A.1 Exemplary Parameter Space

Table A - 1: Exemplary parameter space.

| Representation layer (Bagschik et al., 2018) | Parameter $p_i$ | Scenario | | | | | | | | | | Influence on functional layer | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | free driving | following | lane change | cut in | cut out | cut through | traffic jam | obstacle | Swiss scenario | equ. class scenario | 0 Inf. Access | 1 Inf. Reception | 2 Inf. Process-ing | 3 Sit. Understanding | 4 Behavior. Decision | 5 Action |
| | | number of discretization steps $k_i$ | | | | | | | | | | | | | | | |
| 1 Road Level | width lane 1 | 2 | | | | | | | | | | x | x | | | x | |
| | width lane 2 | 3 | | | | | | | | | | x | x | | | x | |
| | curvature | 100 | | | | | | | | | | x | x | | | x | |
| | roadsurface | 10 | | | | | | | | | | x | x | x | x | x | x |
| | slope | 10 | | | | | | | | | | x | x | | x | x | x |
| | type of marking | 6 | | | | | | | | | | x | x | x | x | | |
| 2 Traffic Infrastructure | type of boundaries/ barriers | 10 | | | | | | | | | | x | x | x | | | |
| | speed limit | 5 | | | | | | | | | | | | | x | x | |
| | type of traffic sign | 3 | | | | | | | | | | x | x | x | | | |
| 4 Objects | initial ego speed | 10 | 10 | 10 | - | - | - | 10 | 10 | 10 | 10 | | x | | x | x | x |
| | type of object 1 | - | 5 | 5 | 5 | 5 | 5 | 5 | - | 5 | 5 | x | x | x | x | | |
| | type of object 2 | - | - | - | 5 | 5 | 5 | 5 | - | 5 | 5 | x | x | x | x | | |
| | type of object 3 | - | - | - | - | - | - | - | - | 5 | 5 | | | | | | |
| | type of obstacle | - | - | - | - | - | - | - | 5 | 5 | 5 | x | x | x | x | | |
| | initial speed of object 1 | - | 10 | 10 | - | - | - | - | - | 10 | 10 | | x | | x | x | |
| | initial speed of object 2 | - | - | - | 10 | - | 10 | - | - | 10 | 10 | | x | | x | x | |
| | initial speed of object 3 | - | - | - | - | - | - | - | - | 10 | 10 | | | | | | |
| | final speed of object 2 | - | - | - | 10 | - | - | - | - | - | 10 | x | x | | x | x | |
| | initial group speed ego+obj1 | - | - | - | 10 | - | 10 | - | - | - | 10 | | x | | x | x | x |
| | initial group speed ego+obj1+obj2 | - | - | - | - | 10 | - | - | - | - | 10 | | x | | x | x | x |
| | initial group speed group1 | - | - | - | - | - | - | 15 | - | - | 15 | x | x | | x | x | x |
| | initial distance ego-obj1 | - | 20 | 20 | 20 | 20 | 20 | - | - | 20 | 20 | x | x | | x | x | |
| | initial distance ego-obj2 | - | - | - | 20 | 20 | 20 | - | - | 20 | 20 | x | x | | x | x | |
| | initial distance obj3-obstacle | - | - | - | - | - | - | - | - | 10 | 10 | | | | | | |
| | initial distance ego-group1 | - | - | - | - | - | - | 20 | - | - | 20 | x | x | | x | x | |
| | initial distance ego-obstacle | - | - | - | - | - | - | - | 20 | 10 | 20 | x | x | | x | x | |
| | lateral position of object 1 | - | - | - | - | - | - | - | - | 5 | 5 | x | x | | x | x | |
| | lateral position of object 2 | - | - | - | - | - | - | - | - | 5 | 5 | x | x | | x | x | |
| | lateral position of object 3 | - | - | - | - | - | - | - | - | 5 | 5 | x | x | | x | x | |
| | lateral position of obstacle | - | - | - | - | - | - | - | 15 | 5 | 15 | x | x | | x | x | |
| | manoever of object 1 | - | - | - | - | - | - | - | - | 5 | 5 | x | x | | x | x | |
| | manoever of object 2 | - | - | - | - | - | - | - | - | 5 | 5 | x | x | | x | x | |
| | manoever of object 3 | - | - | - | - | - | - | - | - | 5 | 5 | x | x | | x | x | |
| | cut-in distance | - | - | - | 20 | - | 20 | - | - | - | 20 | x | x | | x | x | |
| | cut-in time | - | - | - | 10 | - | 10 | - | - | - | 10 | x | x | | x | x | |
| | cut-out time | - | - | - | - | 10 | 10 | - | - | - | 10 | x | x | | x | x | |
| 5 Environment | sun position | 250 | | | | | | | | | | | x | x | | | |
| | precipitation (rain, snow, etc.) | 10 | | | | | | | | | | x | x | x | | | x |
| | cloudinees | 5 | | | | | | | | | | x | x | | | | |
| | wind | 20 | | | | | | | | | | | | | | | x |
| | temperature | 30 | | | | | | | | | | | x | | | | x |

Table A - 2: Sizes of the resulting test suites

| **Size of the resulting test suites** | | **Scenario** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | free driving | follow-ing | lane change | cut in | cut out | cut through | traffic jam | obstacle | Swiss scenario |
| system test | $S_{N,\ sys}$ | $4{\cdot}10^{15}$ | $4{\cdot}10^{18}$ | $4{\cdot}10^{18}$ | $8{\cdot}10^{23}$ | $4{\cdot}10^{20}$ | $8{\cdot}10^{23}$ | $3{\cdot}10^{19}$ | $6{\cdot}10^{18}$ | $8{\cdot}10^{30}$ |
| | $S_{10,\ sys}$ | $9{\cdot}10^{12}$ | $3{\cdot}10^{13}$ | $3{\cdot}10^{13}$ | $10^{14}$ | $6{\cdot}10^{13}$ | $10^{14}$ | $5{\cdot}10^{13}$ | $5{\cdot}10^{13}$ | $2{\cdot}10^{13}$ |
| | $S_{3,\ sys}$ | $8{\cdot}10^{5}$ | $8{\cdot}10^{5}$ | $8{\cdot}10^{5}$ | $8{\cdot}10^{5}$ | $8{\cdot}10^{5}$ | $8{\cdot}10^{5}$ | $8{\cdot}10^{5}$ | $8{\cdot}10^{5}$ | $8{\cdot}10^{5}$ |
| particulate testing | $S_{N,\ part}$ | $4{\cdot}10^{13}$ | $4{\cdot}10^{16}$ | $4{\cdot}10^{16}$ | $8{\cdot}10^{21}$ | $4{\cdot}10^{18}$ | $8{\cdot}10^{21}$ | $3{\cdot}10^{17}$ | $6{\cdot}10^{16}$ | $8{\cdot}10^{28}$ |
| | $S_{10,\ part}$ | $2{\cdot}10^{12}$ | $2{\cdot}10^{13}$ | $2{\cdot}10^{13}$ | $6{\cdot}10^{13}$ | $3{\cdot}10^{13}$ | $6{\cdot}10^{13}$ | $2{\cdot}10^{13}$ | $2{\cdot}10^{13}$ | $8{\cdot}10^{12}$ |
| | $S_{3,\ part}$ | $8{\cdot}10^{5}$ | $8{\cdot}10^{5}$ | $8{\cdot}10^{5}$ | $8{\cdot}10^{5}$ | $8{\cdot}10^{5}$ | $8{\cdot}10^{5}$ | $8{\cdot}10^{5}$ | $8{\cdot}10^{5}$ | $8{\cdot}10^{5}$ |

# List of References

**Abbas, H. et al.: A Driver's License Test for Driverless Vehicles (2017)**
Abbas, Houssam; O'Kelly, Matthew E.; Rodionova, Alena; Mangharam, Rahul: A Driver's License Test for Driverless Vehicles, in: Mechanical Engineering Magazine Select Articles (12), Issues 139, S13-S16, 2017

**Abbas, H. et al.: Safe at any speed: A simulation-based test harness for autonomous vehicles (2017)**
Abbas, Houssam; O'Kelly, Matthew; Rodionova, Alena; Mangharam, Rahul: Safe at any speed: A simulation-based test harness for autonomous vehicles, in: International Workshop on Design, Modeling, and Evaluation of Cyber Physical Systems, 2017

**Ahmed, B. S.; Zamli, K. Z.: A review of covering arrays and their application to software testing (2011)**
Ahmed, Bestoun S.; Zamli, Kamal Z.: A review of covering arrays and their application to software testing, in: Journal of Computer Science (9), Issues 7, p. 1375, 2011

**Althoff, M.; Dolan, J. M.: Online verification of automated road vehicles using reachability analysis (2014)**
Althoff, Matthias; Dolan, John M.: Online verification of automated road vehicles using reachability analysis, in: IEEE Transactions on Robotics (4), Issues 30, pp. 903–918, 2014

**Althoff, M.; Lutz, S.: Automatic generation of safety-critical test scenarios (2018)**
Althoff, Matthias; Lutz, Sebastian: Automatic generation of safety-critical test scenarios for collision avoidance of road vehicles, in: 2018 IEEE Intelligent Vehicles Symposium (IV), 2018

**Amersbach, C.; Winner, H.: Functional Decomposition to Reduce Approval Effort for HAD (2017)**
Amersbach, Christian; Winner, Hermann: Functional Decomposition: An Approach to Reduce the Approval Effort for Highly Automated Driving, in: 8. Tagung Fahrerassistenz, München, 2017

**Amersbach, C.; Winner, H.: Test Coverage for Scenario-Based Validation of HAV (2019)**
Amersbach, Christian; Winner, Hermann: Defining Required and Feasible Test Coverage for Scenario-Based Validation of Highly Automated Vehicles (accepted), in: 22nd IEEE Intelligent Transportation Systems Conference (ITSC), © 2019 IEEE, 2019

**Amersbach, C.; Winner, H.: Functional Decomposition to Overcome the Parameter Space Explosion (2019)**
Amersbach, Christian; Winner, Hermann: Functional Decomposition - a Contribution to Overcome the Parameter Space Explosion during Validation of Highly Automated Driving, in: Traffic Injury Prevention sup1, Issues 20, pp. 52–57, 2019

**Anderson, J. M. et al.: Autonomous vehicle technology (2016)**
Anderson, James M.; Nidhi, Kalra; Stanley, Karlyn D.; Sorensen, Paul; Samaras, Constantine; Oluwatola, Oluwatobi A.: Autonomous vehicle technology: A guide for policymakers, Rand Corporation, 2016

**Åsljung, D. et al.: Using EVT for vehicle level safety validation and implications for AV (2017)**
Åsljung, Daniel; Nilsson, Jonas; Fredriksson, Jonas: Using extreme value theory for vehicle level safety validation and implications for autonomous vehicles, in: IEEE Transactions on Intelligent Vehicles (4), Issues 2, pp. 288–297, 2017

**Audi MediaInfo: Automated driving at a new level: the Audi AI traffic jam pilot (2017)**
Audi MediaInfo: Automated driving at a new level: the Audi AI traffic jam pilot; https://www.audi-mediacenter.com/en/press-releases/automated-driving-at-a-new-level-the-audi-ai-traffic-jam-pilot-9300, 2017, Access 13.03.2018

**AVL LIST GMBH: About the project – Enable S3 (2019)**
AVL LIST GMBH: About the project – Enable S3; https://www.enable-s3.eu/about-project/, 2019, Access 04.03.2019

**Baake, U. et al.: Versuchs-und simulationsbasierte Absicherung von ESP-Systemen für Transporter (2014)**
Baake, Uwe; Wüst, Klaus; Maurer, Markus; Lutz, Albert: Versuchs-und simulationsbasierte Absicherung von ESP-Systemen für Transporter, in: ATZ-Automobiltechnische Zeitschrift (2), Issues 116, pp. 46–51, 2014

**Bach, J. et al.: Test scenario selection for system-level verification and validation (2017)**
Bach, Johannes; Langner, Jacob; Otten, Stefan; Sax, Eric; Holzapfel, Marc: Test scenario selection for system-level verification and validation of geolocation-dependent automotive control systems, in: ICE/ITMC, 2017

**Bagschik, G. et al.: […] Architecture Framework for Safe Automated Vehicles (2018)**
Bagschik, Gerrit; Nolte, Marcus; Ernst, Susanne; Maurer, Markus: A System's Perspective Towards an Architecture Framework for Safe Automated Vehicles, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018

**Bagschik, G. et al.: Ontology Based Scene Creation for the Development of Automated Vehicles (2018)**
Bagschik, Gerrit; Menzel, Till; Maurer, Markus: Ontology Based Scene Creation for the Development of Automated Vehicles, in: 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, Suzhou, China, 2018

**BASt: Fahrleistung von Kraftfahrzeugen auf Autobahnen 2017 (2019)**
Bundesanstalt für Straßenwesen: Entwicklung der gesamten Fahrleistung von Kraftfahrzeugen auf Autobahnen in Deutschland von 1990 bis 2017; https://de.statista.com/statistik/daten/studie/155732/umfrage/fahrleistung-auf-autobahnen-in-deutschland/, 2019, Access 20.03.2019

**Bickel, J.: Mth., Identifikation, Diskretisierung und Zuordnung von Einflussparametern (2019)**
Bickel, Julia: Entwicklung einer Methode zur systematischen Identifikation, Diskretisierung und Zuordnung von Einflussparametern für die Validierung hochautomatisierter Fahrfunktionen, Mth.
TU Darmstadt, Fachgebiet Fahrzeugtechnik, Darmstadt, 2019

**BMVI: Digital Test Beds (2019)**
BMVI: Digital Test Beds; https://www.bmvi.de/EN/Topics/Digital-Matters/Digital-Test-Beds/digital-test-beds.html, 2019, Access 17.05.2019

**Böde, E. et al.: Efficient Splitting of Test and Simulation Cases (2018)**
Böde, Eckard; Büker, Matthias; Eberle, Ulrich; Fränzle, Martin; Gerwinn, Sebastian; Kramer, Birte: Efficient Splitting of Test and Simulation Cases for the Verification of Highly Automated Driving Functions, in: Gallina, Barbara; Skavhaug, Amund; Bitsch, Friedemann (Eds.): Computer Safety, Reliability, and Security, Springer International Publishing, Cham, 2018

**Büker, M. et al.: Identifikation von Automationsrisiken hochautomatisierter Fahrfunktionen (2019)**
Büker, Matthias; Kramer, Birte; Böde, Eckard; Vander Maelen, Sebastian; Fränzle, Martin: Identifikation von Automationsrisiken hochautomatisierter Fahrfunktionen in PEGASUS, in: AAET Automatisiertes und vernetztes Fahren, pp. 315–329, 2019

**Cao, P. et al.: Perception sensor modeling for virtual validation of automated driving (2015)**
Cao, Peng; Wachenfeld, Walther; Winner, Hermann: Perception sensor modeling for virtual validation of automated driving, in: it-Information Technology (4), Issues 57, pp. 243–251, 2015

**Chen, W.; Kloul, L.: Ontology-based Approach to Generate ADAS Use Cases of Highway Traffic (2018)**
Chen, Wei; Kloul, Leila: An Ontology-based Approach to Generate the Advanced Driver Assistance Use Cases of Highway Traffic, in: 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 2018

**Christiansen, M.: In geheimer Mission: auf Abnahmefahrt mit der neuen Mercedes E-Klasse, W213 (2015)**
Christiansen, Marc: In geheimer Mission: auf Abnahmefahrt mit der neuen Mercedes E-Klasse, W213; http://5komma6.mercedes-benz-passion.com/in-geheimer-mission-auf-abnahmefahrt-mit-der-neuen-mercedes-e-klasse-w213/, 2015, Access 12.07.2017

**Cohen, G.: Optimization by decomposition and coordination: A unified approach (1978)**
Cohen, Guy: Optimization by decomposition and coordination: A unified approach, in: IEEE Transactions on automatic control (2), Issues 23, pp. 222–232, 1978

**Cotter, S. et al.: The institutional context for ADAS: A code of practice for development (2006)**
Cotter, S.; Hopkin, J.; Stevens, A.; Burrows, A.; Kompfner, P.; Flanment, M.: The institutional context for advanced driver assistance systems: A code of practice for development, in: PROCEEDINGS OF THE 13th ITS WORLD CONGRESS, LONDON, 8-12 OCTOBER 2006, 2006

**Dantzig, G. B.; Wolfe, P.: Decomposition principle for linear programs (1960)**
Dantzig, George B.; Wolfe, Philip: Decomposition principle for linear programs, in: Operations research (1), Issues 8, pp. 101–111, 1960

**Destatis: Verkehrsunfälle - Zeitreihen 2017 (2018)**
Statistisches Bundesamt: Verkehrsunfälle - Zeitreihen 2017, 2018

**Dietmayer, K. C. et al.: Representation of fused environment data (2016)**
Dietmayer, Klaus C. J.; Reuter, Stephan; Nuss, Dominik: Representation of fused environment data, in: Winner, Hermann et al. (Eds.): Handbook of Driver Assistance Systems, Springer International Publishing, Cham, 2016

**DLR: About PEGASUS - project homepage (2019)**
German Aerospace Center: About PEGASUS - project homepage; https://www.pegasusprojekt.de/en/about-PEGASUS, 2019, Access 23.05.2019

**Donges, E.: Aspekte der aktiven Sicherheit bei der Führung von Personenkraftwagen (1982)**
Donges, Edmund: Aspekte der aktiven Sicherheit bei der Führung von Personenkraftwagen, in: Automob-Ind (2), Issues 27, 1982

**Donges, E.: Vorhersehbarkeit als Auslegungskonzept für Maßnahmen zur aktiven Sicherheit (1992)**
Donges, E.: Das Prinzip Vorhersehbarkeit als Auslegungskonzept für Maßnahmen zur aktiven Sicherheit im Straßenverkehr, in: Das Mensch-Maschine System im Verkehr, VDI-Berichte, Issues 948, 1992

**Donges, E.: Driver Behavior Models (2016)**
Donges, Edmund: Driver Behavior Models, in: Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort, pp. 19–33, 2016

**Endsley, M. R.: Toward a theory of situation awareness in dynamic systems (1995)**
Endsley, Mica R.: Toward a theory of situation awareness in dynamic systems, in: Human factors (1), Issues 37, pp. 32–64, 1995

**FGSV: Richtlinien für die Anlage von Autobahnen (RAA) (2008)**
Forschungsgesellschaft für Straßen- und Verkehrswesen: Richtlinien für die Anlage von Autobahnen (RAA), FGSV-Verl., Köln, 2008

**Gao, F. et al.: A test scenario automatic generation strategy for intelligent driving systems (2019)**
Gao, Feng; Duan, Jianli; He, Yingdong; Wang, Zilong: A test scenario automatic generation strategy for intelligent driving systems, in: Mathematical Problems in Engineering, Issues 2019, 2019

**Gasser, T. M. et al.: Framework Conditions for the Development of DAS (2016)**
Gasser, Tom M.; Seeck, Andre; Smith, Bryant W.: Framework Conditions for the Development of Driver Assistance Systems, in: Winner, Hermann et al. (Eds.): Handbook of Driver Assistance Systems, Springer International Publishing, Cham, 2016

**Gat, E. et al.: On three-layer architectures (1998)**
Gat, Erann; Bonnasso, R. P.; Murphy, Robin; others: On three-layer architectures, in: Artificial intelligence and mobile robots, Issues 195, p. 210, 1998

**GitHub: open_simulation_interface: osi3::GroundTruth Struct Reference (2019)**
GitHub: open_simulation_interface: osi3::GroundTruth Struct Reference; https://opensimulationinterface.github.io/open-simulation-interface/structosi3_1_1GroundTruth.html#details, 2019, Access 29.07.2019

**GitHub: open_simulation_interface: osi3::SensorView Struct Reference (2019)**
GitHub: open_simulation_interface: osi3::SensorView Struct Reference; https://opensimulationinterface.github.io/open-simulation-interface/structosi3_1_1SensorView.html, 2019, Access 29.07.2019

**Graab, B. et al.: Analyse von Verkehrsunfällen […] für die Entwicklung adaptiver FAS (2008)**
Graab, B.; Donner, E.; Chiellino, U.; Hoppe, M.: Analyse von Verkehrsunfällen hinsichtlich unterschiedlicher Fahrerpopulationen und daraus ableitbarer Ergebnisse für die Entwicklung adaptiver Fahrerassistenzsysteme, in: TU München & TÜV Süd Akademie GmbH (Eds.), Conference: Active Safety Through Driver Assistance. München, 2008

**Grindal, M. et al.: Combination testing strategies: a survey (2005)**
Grindal, Mats; Offutt, Jeff; Andler, Sten F.: Combination testing strategies: a survey, in: Software Testing, Verification and Reliability (3), Issues 15, pp. 167–199, 2005

**Gruber, F.; Althoff, M.: Anytime Safety Verification of Autonomous Vehicles (2018)**
Gruber, Felix; Althoff, Matthias: Anytime Safety Verification of Autonomous Vehicles, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018

**Gründl, M.: Diss., Fehler und Fehlverhalten als Ursache von Verkehrsunfällen (2005)**
Gründl, Martin: Fehler und Fehlverhalten als Ursache von Verkehrsunfällen und Konsequenzen für das Unfallvermeidungspotenzial und die Gestaltung von Fahrerassistenzsystemen, Dissertation Universität Regensburg, 2005

**Hacker, W.: Allgemeine Arbeitspsychologie (1998)**
Hacker, W.: Allgemeine Arbeitspsychologie, in: Bern: Huber, 1998

**Hallerbach, S. et al.: Simulation-based identification of critical scenarios (2018)**
Hallerbach, Sven; Xia, Yiqun; Eberle, Ulrich; Koester, Frank: Simulation-based identification of critical scenarios for cooperative and automated vehicles, in: SAE International Journal of Connected and Automated Vehicles 2018-01-1066, Issues1, pp. 93–106, 2018

**Hanke, T. et al.: OSI: A generic interface for the environment perception of AD functions (2017)**
Hanke, Timo; Hirsenkorn, Nils; van Driesten, Carlo; Garcia Ramos, Pilar; Schiementz, Mark; Schneider, Sebastian: Open Simulation Interface: A generic interface for the environment perception of automated driving functions in virtual scenarios; http://www.hot.ei.tum.de/forschung/automotive-veroeffentlichungen/, 2017

**Hohm, A. et al.: Automated driving in real traffic (2014)**
Hohm, Andree; Lotz, Felix; Fochler, Oliver; Lueke, Stefan; Winner, Hermann: Automated driving in real traffic: from current technical approaches towards architectural perspectives, 2014

**Holder, M. et al.: Challenges in Radar Sensor Modeling for Virtual Validation of AD (2018)**
Holder, Martin; Rosenberger, Philipp; Winner, Hermann; D'hondt, Thomas; Makkapati, Vamsi P.; Maier, Michael; Schreiber, Helmut; Magosi, Zoltan; Slavik, Zora; Bringmann, Oliver; others: Measurements revealing Challenges in Radar Sensor Modeling for Virtual Validation of Autonomous Driving, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018

**Hoßbach, P. M.: Masterthesis, Automatic Derivation of Dependency Chains (2019)**
Hoßbach, Phillip M.: Automatic Derivation of Dependency Chains within Systems for Automated Driving via Ontology Based Scenario Representations, Masterthesis TU Darmstadt, Fachgebiet Fahrzeugtechnik, Darmstadt, 2019

**Huang, Z. et al.: Evaluation of automated vehicles in the frontal cut-in scenario (2017)**
Huang, Zhiyuan; Zhao, Ding; Lam, Henry; LeBlanc, David J.; Peng, Huei: Evaluation of automated vehicles in the frontal cut-in scenario—an enhanced approach using piecewise mixture models, in: 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017

**Huang, Z. et al.: Sequential experimentation to efficiently test automated vehicles (2017)**
Huang, Zhiyuan; Lam, Henry; Zhao, Ding: Sequential experimentation to efficiently test automated vehicles, in: 2017 Winter Simulation Conference (WSC), 2017

**Huang, Z. et al.: Towards affordable on-track testing for autonomous vehicle (2017)**
Huang, Zhiyuan; Lam, Henry; Zhao, Ding: Towards affordable on-track testing for autonomous vehicle—A Kriging-based statistical approach, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 2017

**Huelsen, M.: Diss., Knowledge-based driver assistance systems (2014)**
Huelsen, Michael: Knowledge-based driver assistance systems: traffic situation description and situation feature relevance, Dissertation Karlsruhe Institut of Technology (KIT), Springer, 2014

**IEEE: IEEE 610: Standard Glossary of Software Engineering Terminology (1990)**
IEEE: IEEE 610: Standard Glossary of Software Engineering Terminology, IEEE, Piscataway, NJ, USA, 1990

**ISO: ISO/TC 22/SC 31 - Data communication**
International Organization for Standardization: ISO/TC 22/SC 31 - Data communication; https://www.iso.org/committee/5383568.html, Access 29.07.2019

**ISO: ISO 26262: Road vehicles – Functional safety (2018)**
International Organization for Standardization: , ISO ISO 26262: Road vehicles – Functional safety, International Organization for Standardization, 2018

**ISO: ISO/PAS 21448:2019: SOTIF (2019)**
International Organization for Standardization: , PAS ISO/PAS 21448:2019 Road vehicles - Safety of the intended functionality, International Organization for Standardization, 2019

**Junietz, P. et al.: Criticality Metric for the Safety Validation of Automated Driving (2018)**
Junietz, Philipp; Bonakdar, Farid; Klamann, Björn; Winner, Hermann: Criticality Metric for the Safety Validation of Automated Driving using Model Predictive Trajectory Optimization, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018

**Junietz, P. et al.: Evaluation of Different Approaches to Address Safety Validation of AD (2018)**
Junietz, Philipp; Wachenfeld, Walther; Klonecki, Kamil; Winner, Hermann: Evaluation of Different Approaches to Address Safety Validation of Automated Driving, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018

**Junietz, P. et al.: The Risk-Free VAAFO Tool (2019)**
Junietz, Philipp; Wachenfeld, Walther; Schönemann, Valerij; Domhardt, Kai; Tribelhorn, Wadim; Winner, Hermann: Gaining Knowledge on Automated Driving's Safety—The Risk-Free VAAFO Tool, in: Control Strategies for Advanced Driver Assistance Systems and Autonomous Driving Functions, Springer, 2019

**Junietz, P. et al.: Macroscopic Safety Requirements for HAD (2019)**
Junietz, Philipp; Steininger, Udo; Winner, Hermann: Macroscopic Safety Requirements for Highly Automated Driving, in: Transportation Research Record, 2019

**Junietz, P.: Diss., Microscopic and Macroscopic Risk Metrics (Announced for 2019)**
Junietz, Philipp: Microscopic and Macroscopic Risk Metrics for the Safety Validation of Automated Driving., Dissertation TU Darmstadt, Announced for 2019

**Kalra, N.: Challenges and Approaches to Realizing AV Safety (2017)**
Kalra, Nidhi: Challenges and Approaches to Realizing Autonomous Vehicle Safety, RAND, 2017

**Kalra, N.; Paddock, S. M.: Driving to Safety: How Many Miles? (2016)**
Kalra, Nidhi; Paddock, Susan M.: Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?, 2016

**Kamali, M. et al.: Formal verification of autonomous vehicle platooning (2017)**
Kamali, Maryam; Dennis, Louise A.; McAree, Owen; Fisher, Michael; Veres, Sandor M.: Formal verification of autonomous vehicle platooning, in: Science of Computer Programming, Issues 148, pp. 88–106, 2017

**Klamann, B. et al.: Defining Pass-/Fail-Criteria (2019)**
Klamann, Björn; Lippert, Moritz; Amersbach, Christian; Winner, Hermann: Defining Pass-/Fail-Criteria for Particular Tests of Automated Driving Functions (accepted), in: 22nd IEEE Intelligent Transportation Systems Conference (ITSC), © 2019 IEEE, 2019

**Koenig, A. et al.: Bridging the gap between open loop tests and statistical validation for HAD (2017)**
Koenig, Alexander; Gutbrod, Michael; Hohmann, Sören; Ludwig, Julian: Bridging the gap between open loop tests and statistical validation for highly automated driving, in: SAE International journal of transportation safety (1), Issues 5, pp. 81–87, 2017

**Koenig, A. et al.: Passive HAD as a concept for validating HAV (2018)**
Koenig, Alexander; Witzlsperger, Kathrin; Leutwiler, Florin; Hohmann, Sören: Overview of HAD validation and passive HAD as a concept for validating highly automated cars, in: at-Automatisierungstechnik (2), Issues 66, pp. 132–145, 2018

**Koren, M. et al.: Adaptive stress testing for autonomous vehicles (2018)**
Koren, Mark; Alsaif, Saud; Lee, Ritchie; Kochenderfer, Mykel J.: Adaptive stress testing for autonomous vehicles, in: 2018 IEEE Intelligent Vehicles Symposium (IV), 2018

**Krajewski, R. et al.: The highD Dataset (2018)**
Krajewski, Robert; Bock, Julian; Kloeker, Laurent; Eckstein, Lutz: The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems, in: 2018 IEEE 21st International Conference on Intelligent Transportation Systems (ITSC), 2018

**Kuhn, D. R. et al.: Software fault interactions and implications for software testing (2004)**
Kuhn, D. R.; Wallace, Dolores R.; Gallo, Albert M.: Software fault interactions and implications for software testing, in: IEEE transactions on software engineering (6), Issues 30, pp. 418–421, 2004

**Kuhn, D. R. et al.: Practical combinatorial testing (2010)**
Kuhn, D. R.; Kacker, Raghu N.; Lei, Yu: Practical combinatorial testing, in: NIST special Publication (142), Issues 800, p. 142, 2010

**Lambert, F.: Tesla Model S driver crashes into a van while on Autopilot [Video] (2016)**
Lambert, Fredric: Tesla Model S driver crashes into a van while on Autopilot [Video]; https://electrek.co/2016/05/26/tesla-model-s-crash-autopilot-video/, 2016, Access 31.08.2017

**Langner, J. et al.: Estimating the Uniqueness of Test Scenarios using Autoencoders (2018)**
Langner, Jacob; Bach, Johannes; Ries, Lennart; Otten, Stefan; Holzäpfel, Marc; Sax, Eric: Estimating the Uniqueness of Test Scenarios derived from Recorded Real-World-Driving-Data using Autoencoders, in: 2018 IEEE Intelligent Vehicles Symposium (IV), 2018

**Lei, Y. et al.: IPOG/IPOG-D: efficient test generation for multi-way combinatorial testing (2008)**
Lei, Yu; Kacker, Raghu; Kuhn, D. R.; Okun, Vadim; Lawrence, James: IPOG/IPOG-D: efficient test generation for multi-way combinatorial testing, in: Software Testing, Verification and Reliability (3), Issues 18, pp. 125–148, 2008

**Leveson, N.: An STPA Primer (2013)**
Leveson, Nancy: An STPA Primer, 2013

**Liu, P. et al.: How Safe Is Safe Enough for Self-Driving Vehicles? (2018)**
Liu, Peng; Yang, Run; Xu, Zhigang: How Safe Is Safe Enough for Self-Driving Vehicles?, in: Risk analysis, 2018

**Lotz, F. G.: Diss., Referenzarchitektur für die Fahrzeugführung (2017)**
Lotz, Felix G. O.: Eine Referenzarchitektur für die assistierte und automatisierte Fahrzeugführung mit Fahrereinbindung, Dissertation TU Darmstadt, 2017

**Lu, Y.: Masterthesis, Trajektorienplanung und Fehlerursachenanalyse in der Simulation (2015)**
Lu, Yongling: Trajektorienplanung und Fehlerursachenanalyse für die automatisierte Autobahnfahrt in der Simulation, Masterthesis
TU Darmstadt, Fachgebiet Fahrzeugtechnik, 2015

**Manzinger, S. et al.: Kooperative Bewegungsplanung autonomer Fahrzeuge (2017)**
Manzinger, Stefanie; Leibold, Marion; Althoff, Matthias: Kooperative Bewegungsplanung autonomer Fahrzeuge unter Verwendung von Manöver-Templates, in: AAET-Automatisiertes und vernetztes Fahren, 2017

**Matthaei, R.: Diss., Wahrnehmungsgestützte Lokalisierung (2015)**
Matthaei, Richard: Wahrnehmungsgestützte Lokalisierung in fahrstreifengenauen Karten für Assistenzsysteme und automatisches Fahren in urbaner Umgebung, Dissertation TU Braunschweig, Shaker Verlag, 2015

**Matthaei, R.; Maurer, M.: Functional System Architecture (2018)**
Matthaei, Richard; Maurer, Markus: Functional System Architecture for an Autonomous on-Road Motor Vehicle, in: Automotive Systems Engineering II, Springer, 2018

**Menzel, T. et al.: Scenarios for development, test and validation of automated vehicles (2018)**
Menzel, Till; Bagschik, Gerrit; Maurer, Markus: Scenarios for development, test and validation of automated vehicles, in: 2018 IEEE Intelligent Vehicles Symposium (IV), 2018

**Menzel, T. et al.: From Functional to Logical Scenarios (2019)**
Menzel, Till; Bagschik, Gerrit; Isensee, Leon; Schomburg, Andre; Maurer, Markus: From Functional to Logical Scenarios: Detailing a Keyword-Based Scenario Description for Execution in a Simulation Environment, in: CoRRabs/1905.03989, 2019

**Mitsch, S. et al.: On provably safe obstacle avoidance for autonomous robotic ground vehicles (2013)**
Mitsch, Stefan; Ghorbal, Khalil; Platzer, André: On provably safe obstacle avoidance for autonomous robotic ground vehicles, in: Robotics: Science and Systems IX, Technische Universität Berlin, Berlin, Germany, June 24-June 28, 2013, 2013

**Nie, C.; Leung, H.: A survey of combinatorial testing (2011)**
Nie, Changhai; Leung, Hareton: A survey of combinatorial testing, in: ACM Computing Surveys (CSUR) (2), Issues 43, p. 11, 2011

**Nolte, M. et al.: Skill- and ability-based development process (2017)**
Nolte, Marcus; Bagschik, Gerrit; Jatzkowski, Inga; Stolte, Torben; Reschka, Andreas; Maurer, Markus: Towards a skill- and ability-based development process for self-aware automated road vehicles, in: IEEE ITSC 2017, Yokohama, IEEE, Piscataway, NJ, 2017

**PEGASUS Project Office: Description of the PEGASUS-Method (2019)**
PEGASUS Project Office: Description of the PEGASUS-Method; https://www.pegasusprojekt.de/en/pegasus-method, 2019, Access 17.07.2019

**PEGASUS Project Office: PEGASUS Abschlussveranstaltung 2019 - The Highway-Chauffeur (2019)**
PEGASUS Project Office: PEGASUS Abschlussveranstaltung 2019 - The Highway-Chauffeur; https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/04_The_Highway_Chauffeur.pdf, 2019, Access 02.08.2019

**PEGASUS Project Office: PEGASUS METHOD (2019)**
PEGASUS Project Office: PEGASUS METHOD; https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/PEGASUS-Gesamtmethode.pdf, 2019, Access 10.06.2019

**Ponn, T. et al.: Identify Relevant Scenarios for Type Approval of AV (2019)**
Ponn, Thomas; Diermmeyer, Frank; Gnandt, Christian: An Optimization-Based Method to Identify Relevant Scenarios for Type Approval of Automated Vehicles, in: 26th International Technical Conference and exhibition on the Enhanced Safety of Vehicles (ESV), Eindhoven, 2019

**Pütz, A. et al.: Database approach for the sign-off process of highly automated vehicles (2017)**
Pütz, Andreas; Zlocki, Adrian; Küfen, Jörg; Bock, Julian; Eckstein, Lutz: Database approach for the sign-off process of highly automated vehicles, in: 25th International Technical Conference on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety Administration, 2017

**Pütz, A. et al.: System validation of HAV with a database of relevant traffic scenarios (2017)**
Pütz, Andreas; Zlocki, Adrian; Bock, Julian; Eckstein, Lutz: System validation of highly automated vehicles with a database of relevant traffic scenarios, in: 12th ITS European Congress, Strasbourg, 2017

**Rasmussen, J.: Human errors (1982)**
Rasmussen, Jens: Human errors. A taxonomy for describing human malfunction in industrial installations, in: Journal of occupational accidents 2-4, Issues 4, pp. 311–333, 1982

**Rasmussen, J.: Skills, rules, and knowledge […] in human performance models (1983)**
Rasmussen, Jens: Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models, in: IEEE transactions on systems, man, and cybernetics (3), pp. 257–266, 1983

**Rathgeber, C.: Diss., Trajektorienplanung und-folgeregelung (2016)**
Rathgeber, Christian: Trajektorienplanung und-folgeregelung für assistiertes bis hochautomatisiertes Fahren, Dissertation TU Berlin, 2016

**Reason, J.: Human error (1990)**
Reason, James: Human error, Cambridge university press, 1990

**Reason, J.: The Contribution of Latent Human Failures to the Breakdown of Complex Systems (1990)**
Reason, J.: The Contribution of Latent Human Failures to the Breakdown of Complex Systems, in: Philosophical Transactions of the Royal Society B: Biological Sciences (1241), Issues 327, pp. 475–484, 1990

**Reschka, A. et al.: Ability and skill graphs (2015)**
Reschka, Andreas; Bagschik, Gerrit; Ulbrich, Simon; Nolte, Marcus; Maurer, Markus: Ability and skill graphs for system modeling, online monitoring, and decision support for vehicle guidance systems, in: Intelligent Vehicles Symposium (IV), 2015 IEEE, 2015

**Reschka, A.: Diss., Fertigkeiten-und Fähigkeitengraphen (2017)**
Reschka, Andreas: Fertigkeiten-und Fähigkeitengraphen als Grundlage des sicheren Betriebs von automatisierten Fahrzeugen im öffentlichen Straßenverkehr in städtischer Umgebung, Dissertation TU Braunschweig, 2017

**Rosenberger, P. et al.: Towards a Validation Methodology for Sensor Models (2019)**
Rosenberger, Philipp; Wendler, Jan T.; Holder, Martin; Linnhoff, Clemens; Berghöfer, Moritz; Winner, Hermann; Maurer, Markus: Towards a Generally Accepted Validation Methodology for Sensor Models - Challenges, Metrics, and First Results, in: Grazer Symposium Virtuelles Fahrzeug, 2019

**Saaty, T. L.: The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation (1980)**
Saaty, T. L.: The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation, Advanced book program, McGraw-Hill, 1980

**SAE: Cybersecurity Guidebook for Cyber-Physical Vehicle Systems (2016)**
Society of Automotive Engineers: Cybersecurity Guidebook for Cyber-Physical Vehicle Systems, 2016

**SAE: J3016: Taxonomy and Definitions for […] Vehicle Automated Driving Systems (2018)**
Society of Automotive Engineers: J3016: Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems, 2018

**Sage, A.; Lienert, P.: GM plans large-scale launch of self-driving cars in U.S. cities in 2019 (2017)**
Sage, Alexandria; Lienert, Paul: GM plans large-scale launch of self-driving cars in U.S. cities in 2019; https://www.reuters.com/article/us-gm-autonomous/gm-plans-large-scale-launch-of-self-driving-cars-in-u-s-cities-in-2019-idUSKBN1DU2H0, 2017, Access 17.04.2018

**Sauerbier, J. et al.: Definition of Scenarios for Safety Validation of Automated Driving Functions (2019)**
Sauerbier, Jan; Bock, Julian; Weber, Hendrik; Eckstein, Lutz: Definition of Scenarios for Safety Validation of Automated Driving Functions, in: ATZ worldwide (1), Issues 121, pp. 42–45, 2019

**Schaller, T.; Dehlink, B.: Sensor standardization initiative for automated driving (2017)**
Schaller, Thomas; Dehlink, Bernhard: Sensor standardization initiative for automated driving, 8. Tagung Fahrerassistenz, München, 2017

**Schmidt, G.: Diss., Influence of Warnings on Collision Avoidance Behavior (2012)**
Schmidt, Gerald: The Influence of Anticipation and Warnings on Collision Avoidance Behavior of Attentive Drivers, Dissertation Julius Maximilians Universität Würzburg, 2012

**Schoitsch, E. et al.: The need for safety and cyber-security co-engineering and standardization (2016)**
Schoitsch, Erwin; Schmittner, Christoph; Ma, Zhendong; Gruber, Thomas: The need for safety and cyber-security co-engineering and standardization for highly automated automotive vehicles, in: Advanced Microsystems for Automotive Applications 2015, Springer, 2016

**Schuldt, F. et al.: Effiziente systematische Testgenerierung für FAS in virt. Umgebungen (2013)**
Schuldt, Fabian; Saust, Falko; Lichte, Bernd; Maurer, Markus; Scholz, Stephan: Effiziente systematische Testgenerierung für Fahrerassistenzsysteme in virtuellen Umgebungen, in: Automatisierungssysteme, Assistenzsysteme und Eingebettete Systeme Für Transportmittel, 2013

**Schuldt, F. et al.: Zuordnung von Testfällen auf X-in-the-Loop Verfahren (2015)**
Schuldt, Fabian; Menzel, Till; Maurer, Markus: Eine Methode für die Zuordnung von Testfällen für automatisierte Fahrfunktionen auf X-in-the-Loop Verfahren im modularen virtuellen Testbaukasten, in: Workshop Fahrerassistenzsysteme, 2015

**Schuldt, F.: Diss., Ein Beitrag für den methodischen Test von autom. Fahrfunktionen (2017)**
Schuldt, Fabian: Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen-English title: Towards testing of automated driving functions in virtual driving environments, Dissertation TU Braunschweig, 2017

**Schuldt, F. et al.: Efficient, systematic test case generation for ADAS in virtual environments (2018)**
Schuldt, Fabian; Reschka, Andreas; Maurer, Markus: A method for an efficient, systematic test case generation for advanced driver assistance systems in virtual environments, in: Automotive Systems Engineering II, Springer, 2018

**Shalev-Shwartz, S. et al.: On a Formal Model of Safe and Scalable Self-driving Cars (2017)**
Shalev-Shwartz, Shai; Shammah, Shaked; Shashua, Amnon: On a Formal Model of Safe and Scalable Self-driving Cars, 2017

**Sommerville, I.: Software engineering (2006)**
Sommerville, Ian: Software engineering, 8. Edition, New York: Addison-Wesley, 2006

**Steimle, M. et al.: Classifying Test Bench Configurations (2019)**
Steimle, Markus; Menzel, Till; Maurer, Markus: A Method for Classifying Test Bench Configurations in a Scenario-Based Test Approach for Automated Vehicles, 2019

**Stellet, J. E. et al.: Testing of advanced driver assistance towards automated driving (2015)**
Stellet, Jan E.; Zofka, Marc R.; Schumacher, Jan; Schamm, Thomas; Niewels, Frank; Zöllner, J. M.: Testing of advanced driver assistance towards automated driving, in: 2015 IEEE 18th International Conference on Intelligent Transportation Systems, 2015

**Stolte, T. et al.: Safety goals and functional safety requirements (2016)**
Stolte, Torben; Bagschik, Gerrit; Maurer, Markus: Safety goals and functional safety requirements for actuation systems of automated vehicles, in: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, IEEE, 2016

**Stolte, T. et al.: Hazard analysis and risk assessment for an automated unmanned protective vehicle (2017)**
Stolte, Torben; Bagschik, Gerrit; Reschka, Andreas; Maurer, Markus: Hazard analysis and risk assessment for an automated unmanned protective vehicle, in: 2017 IEEE Intelligent Vehicles Symposium (IV), 2017

**Tao, J. et al.: On the Industrial Application of Combinatorial Testing for AD Functions (2019)**
Tao, Jianbo; Li, Yihao; Wotawa, Franz; Felbinger, Hermann; Nica, Mihai: On the Industrial Application of Combinatorial Testing for Autonomous Driving Functions, in: 2019 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW), 2019

**Tatar, M.: Chasing Critical Situations in Large Parameter Spaces (2018)**
Tatar, Mugur: Chasing Critical Situations in Large Parameter Spaces, Autonomous Vehicle Test & Development Symposium, Stuttgart, 2018

**Templeton, B.: Tesla's "Shadow" Testing (2019)**
Templeton, Brad: Tesla's "Shadow" Testing Offers A Useful Advantage On The Biggest Problem In Robocars; https://www.forbes.com/sites/bradtempleton/2019/04/29/teslas-shadow-testing-offers-a-useful-advantage-on-the-biggest-problem-in-robocars/#2da69f933c06, 2019, Access 24.06.2019

**Toselli, A.; Widlund, O.: Domain decomposition methods-algorithms and theory (2006)**
Toselli, Andrea; Widlund, Olof: Domain decomposition methods-algorithms and theory, Issues 34, Springer Science & Business Media, 2006

**Ulbrich, S. et al.: Defining […] the terms scene, situation, and scenario for automated driving (2015)**
Ulbrich, Simon; Menzel, Till; Reschka, Andreas; Schuldt, Fabian; Maurer, Markus: Defining and substantiating the terms scene, situation, and scenario for automated driving, in: Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on, 2015

**VDA QMC: Automotive SPICE (2017)**
VDA QMC Working Group 13 / Automotive SIG: Automotive SPICE Process Assessment / Reference Model, Version 3.1, 2017

**VDI: 2206: Entwicklungsmethodik für mechatronische Systeme (2004)**
VDI: 2206: Entwicklungsmethodik für mechatronische Systeme, 2004

**VIRES GmbH: OpenDRIVE® / OpenCRG® Product Data Sheet (2011)**
VIRES Simulationstechnologie GmbH: OpenDRIVE® / OpenCRG® Product Data Sheet; http://www.opendrive.org/docs/VIRES_ODR_OCRG.pdf, 2011, Access 22.06.2019

**VIRES GmbH: OpenSCENARIO Homepage (2018)**
VIRES Simulationstechnologie GmbH: OpenSCENARIO Homepage; http://www.openscenario.org/, 2018, Access 22.06.2019

**Volkswagen AG: Volkswagen tests highly-automated driving in Hamburg (2019)**
Volkswagen AG: Volkswagen tests highly-automated driving in Hamburg; https://www.volkswagenag.com/en/news/2019/04/volkswagen-tests-highly-automated-driving-in-hamburg.html, 2019, Access 17.05.2019

**Vukadinovic, V. et al.: 3GPP C-V2X and IEEE 802.11 p for V2V communications (2018)**
Vukadinovic, Vladimir; Bakowski, Krzysztof; Marsch, Patrick; Garcia, Ian D.; Xu, Hua; Sybis, Michal; Sroka, Pawel; Wesolowski, Krzysztof; Lister, David; Thibault, Ilaria: 3GPP C-V2X and IEEE 802.11 p for Vehicle-to-Vehicle communications in highway platooning scenarios, in: Ad Hoc Networks, Issues 74, pp. 17–29, 2018

**Wachenfeld, W. et al.: The worst-time-to-collision metric for situation identification (2016)**
Wachenfeld, Walther; Junietz, Philipp; Wenzel, Raphael; Winner, Hermann: The worst-time-to-collision metric for situation identification, in: 2016 IEEE Intelligent Vehicles Symposium (IV), Gotenburg, Sweden, IEEE, Piscataway, NJ, 2016

**Wachenfeld, W.; Winner, H.: VAAFO a new runtime validation method (2015)**
Wachenfeld, Walther; Winner, Hermann: Virtual assessment of automation in field operation a new runtime validation method, in: 10. Workshop Fahrerassistenz-systeme, 2015

**Wachenfeld, W.; Winner, H.: The Release of Autonomous Vehicles (2016)**
Wachenfeld, Walther; Winner, Hermann: The Release of Autonomous Vehicles, in: Winner, Hermann et al. (Eds.): Autonomous Driving, Springer, Berlin, Heidelberg, 2016

**Wang, L. et al.: Prospective safety assessment of HAD functions using stochastic traffic simulation (2017)**
Wang, Lei; Fahrenkrog, Felix; Vogt, Timo; Jung, Olaf; Kates, Ronald: Prospective safety assessment of highly automated driving functions using stochastic traffic simulation, in: 25th International Technical Conference on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety Administration, Detroit, 2017

**WAYMO: On the Road to Fully Self-Driving (2017)**
WAYMO: On the Road to Fully Self-Driving, 2017

**Weigmann, D. A.; Shappell, S. A.: Human factors analysis of postaccident data (1997)**
Weigmann, Douglas A.; Shappell, Scott A.: Human factors analysis of postaccident data: Applying theoretical taxonomies of human error, in: The International Journal of Aviation Psychology (1), Issues 7, pp. 67–81, 1997

**Weitzel, D. A.: Diss., Kontrollierbarkeit nicht situationsgerechter Reaktionen von ADAS (2013)**
Weitzel, Dirk A.: Objektive Bewertung der Kontrollierbarkeit nicht situationsgerechter Reaktionen umfeldsensorbasierter Fahrerassistenzsysteme, Dissertation TU Darmstadt, VDI-Verlag, 2013

**Wickens, C. D.: Engineering psychology and human performance (1992)**
Wickens, Christopher D.: Engineering psychology and human performance, 2. Edition, HarperCollins Publishers, New York, 1992

**Wille, J. M. et al.: Stadtpilot: Driving autonomously on Braunschweig's inner ring road (2010)**
Wille, Jorn M.; Saust, Falko; Maurer, Markus: Stadtpilot: Driving autonomously on Braunschweig's inner ring road, in: 2010 IEEE Intelligent Vehicles Symposium (IV), La Jolla, CA, USA, 2010

**Williams, A. W.; Probert, R. L.: A measure for component interaction test coverage (2001)**
Williams, Alan W.; Probert, Robert L.: A measure for component interaction test coverage, in: Proceedings ACS/IEEE International Conference on Computer Systems and Applications, 2001

**Winner, H.: Patent DE10102771: Einrichtung zum Bereitstellen von Signalen in einem Kraftfahrzeug (2000)**
Winner, Hermann: Patent DE10102771: Einrichtung zum Bereitstellen von Signalen in einem Kraftfahrzeug, Patent DE10102771, 2000

**Winner, H. et al.: Freigabefalle des autonomen Fahrens (2010)**
Winner, H.; Wolf, G.; Weitzel, A.: Freigabefalle des autonomen Fahrens/the approval trap of autonomous driving, in: 15. VDI-Tagung Erprobung und Simulation in der Fahrzeugentwicklung (SIMVEC), Baden-Baden Nr. 2106, VDI-Verl., Düsseldorf, 2010

**Winner, H. et al.: Validation and introduction of automated driving (2018)**
Winner, Hermann; Wachenfeld, Walther; Junietz, Phillip: Validation and introduction of automated driving, in: Automotive Systems Engineering II, Springer, 2018

**Winner, H.; Merkel, N. L.: Mode-Confusion und Inkompatibilitäten (2017)**
Winner, Hermann; Merkel, Nora L.: Mode-Confusion und Inkompatibilitäten in der Migrationsphase des automatisierten Fahrens, in: 8. Darmstädter Kolloquium 7./8. März 2017 Technische Universität Darmstadt Herausgeber: H. Winner und R. Bruder, 2017

**Winner, H.; Wachenfeld, W.: Absicherung automatischen Fahrens (2013)**
Winner, Hermann; Wachenfeld, Walther: Absicherung automatischen Fahrens, 6. Tagung Fahrerassistenz – Der Weg zum automatischen Fahren, München, 2013

**Winner, H.; Wolf, G.: Quo vadis, FAS? (2009)**
Winner, Hermann; Wolf, Gabriele: Quo vadis, FAS?, in: Winner, Hermann; Hakuli, Stephan; Wolf, Gabriele (Eds.): Handbuch Fahrerassistenzsysteme, Vieweg + Teubner, Wiesbaden, 2009

**Xia, Q. et al.: Automatic generation method of test scenario for adas based on complexity (2017)**
Xia, Qin; Duan, Jianli; Gao, Feng; Chen, Tao; Yang, Cai: Automatic generation method of test scenario for adas based on complexity, 2017

**Xia, Q. et al.: Test scenario design for intelligent driving system ensuring coverage and effectiveness (2018)**
Xia, Qin; Duan, Jianli; Gao, Feng; Hu, Qiuxia; He, Yingdong: Test scenario design for intelligent driving system ensuring coverage and effectiveness, in: International Journal of Automotive Technology (4), Issues 19, pp. 751–758, 2018

**Zhao, D. et al.: Accelerated evaluation of automated vehicles in car-following maneuvers (2017)**
Zhao, Ding; Huang, Xianan; Peng, Huei; Lam, Henry; LeBlanc, David J.: Accelerated evaluation of automated vehicles in car-following maneuvers, in: IEEE Transactions on Intelligent Transportation Systems (3), Issues 19, pp. 733–744, 2017

**Zhao, D. et al.: Accelerated Evaluation of Automated Vehicles Safety in Lane-Change Scenarios … (2017)**
Zhao, Ding; Lam, Henry; Peng, Huei; Bao, Shan; LeBlanc, David J.; Nobukawa, Kazutoshi; Pan, Christopher S.: Accelerated Evaluation of Automated Vehicles Safety in Lane-Change Scenarios Based on Importance Sampling Techniques, in: IEEE transactions on intelligent transportation systems : a publication of the IEEE Intelligent Transportation Systems Council (3), Issues 18, pp. 595–607, 2017

**Ziegler, J. et al.: Making Bertha Drive—An Autonomous Journey on a Historic Route (2014)**
Ziegler, Julius; Bender, Philipp; Schreiber, Markus; Lategahn, Henning; Strauss, Tobias; Stiller, Christoph; Dang, Thao; Franke, Uwe; Appenrodt, Nils; Keller, Christoph G.; Kaus, Eberhard; Herrtwich, Ralf G.; Rabe, Clemens; Pfeiffer, David; Lindner, Frank; Stein, Fridtjof; Erbs, Friedrich; Enzweiler, Markus; Knoppel, Carsten; Hipp, Jochen; Haueis, Martin; Trepte, Maximilian; Brenk, Carsten; Tamke, Andreas; Ghanaat, Mohammad; Braun, Markus; Joos, Armin; Fritz, Hans; Mock, Horst; Hein, Martin; Zeeb, Eberhard: Making Bertha Drive—An Autonomous Journey on a Historic Route, in: IEEE Intelligent Transportation Systems Magazine (2), Issues 6, pp. 8–20, 2014

**Zimmer, A.: Wie intelligent darf/muss ein Auto sein? (2001)**
Zimmer, Alf: Wie intelligent darf/muss ein Auto sein? Anmerkungen aus ingenieurpsychologischer Sicht, in: Kraftfahrzeugführung, Springer, 2001

# Own Publications

**Amersbach, C.; Winner, H.:** Defining Required and Feasible Test Coverage for Scenario-Based Validation of Highly Automated Vehicles. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, NZ, 2019 (accepted, pre-print available)

**Amersbach, C.; Winner, H.:** Functional Decomposition - A Contribution to Overcome the Parameter Space Explosion during Validation of Highly Automated Driving In: Traffic Injury Prevention sup1, Issues 20, pp. 52–57, 2019

**Amersbach, C.; Winner, H.:** Functional Decomposition, An Approach to Reduce the Approval Effort for Highly Automated Driving. In: 8. Tagung Fahrerassistenz, 22.-23. November, München, 2017

**Amersbach, C.; Winner, H.:** Funktionale Dekomposition – Ein Beitrag zur Überwindung der Parameterraumexplosion bei der Validation von höher automatisiertem Fahren. In: 12. Workshop Fahrerassistenzsysteme und automatisiertes Fahren, 26.-28. September, Walting, 2018

**Klamann, B.; Lippert M.; Amersbach, C.; Winner, H.:** Defining Pass-/Fail-Criteria for Particular Tests of Automated Driving Functions. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, NZ, 2019 (accepted)

**Klamann, B.; Lippert M.; Amersbach, C.; Winner, H.:** Definition von Bestehens-/ Versagenskriterien für das partikuläre Testen von automatisierten Fahrfunktionen. In: 9. Tagung Automatisiertes Fahren, 21.-22. November, München, 2019 (accepted)

**Woopen, T.; Lampe, B.; Böddeker, T.; Eckstein, L.; Kampmann, A.; Alrifaee, B.; Kowalewski, S.; Moormann, D.; Stolte, T.; Jatzkowski, I.; Maurer, M.; Möstl, M.; Ernst, R.; Ackermann, S.; Amersbach, C.; Leinen, S.; Winner, H.; Püllen, D.; Katzenbeisser, S.; Becker, M.; Stiller, C.; Furmans, K.; Bengler, K.; Diermeyer, F.; Lienkamp, M.; Keilhoff, D.; Reuss, H.-C.; Buchholz, M.; Dietmayer, K.; Lategahn, H.; Siepenkötter, N.; Elbs, M.;v. Hinüber, E.; Dupuis, M.; Hecker, C.:** UNICAR*agil* – Disruptive Modular Architectures for Agile, Automated Vehicle Concepts. In: 27th Aachen Colloquium, 08.-10.October, Aachen, 2018

# Supervised Theses

**Aouini, Rachid:** Analyse des Potentials zur Reduktion des Freigabeaufwandes für hochautomatisiertes Fahren durch funktionale Dekomposition. Masterthesis No. 678/18, 2018.

**Berens, Alexander:** Entwicklung und Implementierung einer Querdynamikregelung für automatisiertes Fahren. Masterthesis No. 652/17, 2018.

**Bickel, Julia:** Entwicklung einer Methode zur systematischen Identifikation, Diskretisierung und Zuordnung von Einflussparametern für die Validierung hochautomatisierter Fahrfunktionen. Masterthesis No. 725/18, 2019.

**Brandau, Michael:** Recherche und Vergleich von Testmethoden und -kriterien für die Informationsaufnahme von Fahrassistenzsystemen und automatisierten Fahrfunktionen. Bachelorthesis No. 1290-B/17, 2017.

**Domharhdt, Kai:** Retrospektive Korrektur von Objektexistenzfehlern in der Umfelderfassung. Bachelorthesis No. 1279/16, 2017.

**Gosewinkel, Dennis:** Entwicklung einer Test- und Absicherungsmethode für ein Formula Student Driverless Fahrzeug. Bachelorthesis No. 1298/17, 2017.

**Hebgen, Niclas:** Definition makroskopischer Sicherheitsanforderungen für automatisiertes Fahren in der Stadt. Masterthesis No. 748/19, 2019.

**Hirsch, Alexander:** Hardwarenahe Implementierung und Erweiterung eines Ansatzes einer Trajektorienplanug für ein vollautomatisiertes Fahrzeug der Formula Student Driverless. Masterthesis No. 653/17, external at Continental Teves AG & Co. OHG, 2018.

**Hofmann, Tobias:** Entwicklung eines Schätzverfahrens für den maximalen Reibwert der Reifen-Straße Paarung eines elektrifizierten Fahrzeugs im automatisierten Fahrbetrieb: ext. Math 681/18, external at Robert Bosch GmbH, 2018.

**Homolla, Tobias:** Konzeptentwicklung für eine 3 DoF Fahrdynamikregelung. Masterthesis No. 680/18, 2018.

**Hoßbach, Phillip:** Ontologie basiertes Systemdesign von vollautomatisierten Fahrzeugen. Masterthesis No. 713/18, external at Daimler AG, 2019.
Available via TU Prints: urn:nbn:de:tuda-tuprints-87490

**Huber, Marc:** Entwicklung einer Prüfmethodik für einen Formula Student Reifen. Bachelorthesis No.1289-A/17, external at Pirelli Deutschland GmbH, 2017.

**Klamann, Björn:** Anforderungsanalyse für Modulschnittstellen zur Absicherung modularer Systeme. Masterthesis No. 673/17, 2018.

**Le Floch, Antoine:** Entwicklung eines Konzeptes zur Implementierung automatisierter Fahrfunktionen in ein Formula Student Electric Fahrzeug. Masterthesis No. 612/16, 2016.

**Lippert, Moritz:** Entwicklung einer Methodik zur Kategorisierung von Streckenabschnitten. Masterthesis No. 679/18, 2018.

**Ludwig, Christian:** Definition von Anforderungen an ein Messdatenformat und an aufzuzeichnende Messgrößen zur Validation automatisierter Fahrfunktionen. Masterthesis No. 749/19, 2019.

**Ludwig, Christian:** Recherche und Vergleich von Ansätzen zur funktionalen Dekomposition von Fahraufgaben. Bachelorthesis No. 1278/16, 2017.

**Pintscher, Patrick:** Entwicklung einer Längs- und Querdynamikregelung für einen Formula-Student-Driverless Rennwagen. Masterthesis No. 630/16, 2017.

**Sarikaya, Erkut:** Entwicklung und Implementation einer Längsdynamikregelung für automatisiertes Fahren. Bachelorthesis No. 1291/17, 2017.

**Scholz, Fabian:** Recherche und Vergleich von Testmethoden und -kriterien für die Informationsverarbeitung und Situationsverständnis von automatisierten Fahrfunktionen. Bachelorthesis No. 1296/17, 2017.

**Tribelhorn, Wadim:** Konzeptionierung und prototypische Implementation einer „Silent Testing"-Methode zur automatisierten Bewertung der Fahrstreifenmarkierungserkennung für automatisierte Fahrzeuge. Masterthesis No. 703/18, external at Continental Teves AG & Co. OHG, 2018.

**Unterschütz, Annemike:** Entwicklung eines IT-Security Konzepts für ein automatisiertes Fahrzeug. Bachelorthesis No. 1324/18, 2018.

**Weber, Nico:** Reduzierung des Parameterraums für die Freigabe von hochautomatisierten Fahrfunktionen. Masterthesis No. 736/19, external at Opel Automobile GmbH, 2019.

**Wende, Simon:** Modellprädiktive Längs- und Querregelung im niedrigen Geschwindigkeitsbereich. Masterthesis No. 682/18, external at Continental Teves AG & Co. OHG, 2018.

**Yu, Zhengdong:** Entwicklung einer Methodik zur Definition von Versagenskriterien zum partikulären Testen hochautomatisierten Fahrfunktionen. Masterthesis No. 696/18, 2018.