Functional Decomposition

An Approach to Reduce the Approval Effort for Highly Automated Driving

Christian Amersbach, Hermann Winner
Institute of Automotive Engineering
Technische Universität Darmstadt
Darmstadt, Germany
{amersbach, winner}@fzd.tu.darmstadt.de

***Abstract*—It is not economically feasible to statistically prove that an automated driving function is safer than a human driver with current test methods such as field operation tests. Therefore, new methods and tools are needed for the release of automated vehicles. The approach of functional decomposition is broadly used in informatics, mathematics and robotics to split complex functions into sub functions. Functional decomposition is also used to analyze human failures that lead to traffic accidents. Assuming that the absence of accidents is the approval criterion for highly automated driving and that accidents can be traced back to failures, the decomposition approach can be used for approval. Within the research project PEGASUS, a scenario-based decomposition approach for approval of the "Autobahn-Chauffer" is developed. This approach is presented here for the first time and proposes a six-layer decomposition of the automated driving function. Based on the functional decomposition and identification of relevant scenarios, particular test cases and corresponding fail criteria can be derived. By eliminating redundant test cases and aggregating test cases that are subsets of each other, the method promises to reduce testing effort.**

Keywords—automated driving; safety assessment; test case generation; decomposition; PEGASUS

I. INTRODUCTION

The technical development of autonomous vehicles is almost finished. 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 [1], the project "Stadtpilot" of TU Braunschweig [2] and many more. However, there is no autonomous vehicle available on the market yet. One reason for that are the high safety requirements for automated driving and the challenge to approve that such a system is safe enough.

This study was founded by the Federal Ministry for Economic Affairs and Energy (BMWi) based on a decision by the German Bundestag within the 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.

A. Current Test Concepts in the Automobile Industry

For driver-only vehicles (i.e. level 0 according to SAE [3]), it is assumed that all components are designed and approved according to industrial standards such as ISO 26262 [4] 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 decades, this has been shown to be a successful proof of safety. For advanced driver assistance systems (ADAS), the Code of Practice assumes that the responsibility for the vehicle's behavior still remains with the human driver that is always in the loop and can overwrite or deactivate the system at any time. Therefore, the results of conducted tests with test drivers can be transferred to future users, similar to driver-only vehicles. [5, 428 ff.]

The standard ISO 26262 [4] is state of the art regarding functional safety for safety-critical E/E systems in road vehicles. All current ADAS are developed according to ISO 26262. However, it cannot be used for the development of highly automated driving (HAD) functions (i.e. level 3 and higher according to SAE [3]) without adaption. 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 permanently monitor the system, this controllability is not given. [6]

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. [7, p. 37], e.g. automatic emergency braking), tests that only focus on the driver's controllability are not sufficient. It must also be proven that the false-positive rate is adequately low. 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 [8], the approval of current level 2 systems requires up to 12 million test kilometers.

B. *Billions of Kilometers until the Release of Vehicles with unsupervised Automated Driving*

If one wants to retain the current test concepts for HAD, the required test distances in real traffic will increase dramatically. Assuming that the automated driving function is twice as safe as a human driver and using the number of fatal accidents as a metric for safety, according to Wachenfeld and Winner [5] around 6.6 billion test kilometers have to be driven in real traffic under representative conditions for a safety assessment of an "Autobahn Chauffeur". Kalra and Paddock [9] calculate required distances up to 11 billion miles for the safety assessment of a level 5 system (according to [3]), based on statistical data from the USA with a similar approach. As can be seen from the mentioned examples, a statistical safety assessment for automated driving functions is not feasible in practice before introduction. Therefore, alternative safety assessment methods are required. For example, so-called ability and skill graphs that are based on ISO 26262 are proposed by Reschka et al. [6] for the development and online monitoring of vehicle guidance systems and are used within the projects "aFAS" and "Stadtpilot".

C. *The research project PEGASUS*

To develop new standards and methods for the approval of automated driving functions, the "**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" (PEGASUS) was launched in 2016. In this joint project, 17 partners from the science and industry field define a state-of-the-art technology for the safeguarding of HAD. The project is promoted by the Federal Ministry for Economic Affairs and Energy (BMWi) and will be finished by the middle of 2019 [10].

D. *Scenario-based approach*

Within the project PEGASUS, a so-called scenario-based approach is applied to reduce the approval effort for highly automated driving. It is assumed that the major part of mileage on the Autobahn goes well without any special events, while critical scenarios are quite rare and randomly distributed in real traffic. Testing of the first mentioned ordinary scenarios is without relevant contribution for the approval process. Therefore, the identification of critical scenarios that can be reproduced in simulation or on test fields should significantly reduce the long driving test distances needed for a statistical approval [11].

II. FUNCTIONAL DECOMPOSITION FOR TEST CASE GENERATION

The approach of decomposing the driving task for test case generation that is presented here for the first time is based on the scenario-based approach and has the potential to reduce the approval effort even more. Within PEGASUS, this new approach is developed for an "Autobahn Chauffeur" as example of use.

A. *The basic concept of functional decomposition*

The concept of functional decomposition is not entirely new. It is used in various domains as for example in robotics, informatics, or accident analysis to segment complex functions or problems into sub functions or sub problems respectively. Functional decomposition is also used to create layer-based, so-called sequential system architectures for ADAS whilst test case generation is not the main reason for decomposing the system [12, 45 ff.].

A five-level decomposition of the human driving task is used by Graab et al. [13] to analyze failures that lead to traffic accidents. Here, the failure chain is decomposed into the following layers:

(1) Information access

(2) Information reception

(3) Information processing

(4) Behavioral decision

(5) Action

As the pass criterion for testing HAD is the absence of accidents and as it is here assumed that accidents can be traced back to failures, this approach can be transferred to test case generation for HAD. In analogy, the driving function is first split into independent functional layers. Hereby not all HAD functions can be split into all of the proposed layers, i.e. if the interfaces between the layers are not accessible, two or more layers need to be combined. In a next step, test cases that are deduced from critical scenarios are decomposed into particular tests, which cover one or more functional layers. This will be handled in detail in sections II D and II E.

B. *Benefits*

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 "driving past 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 a hedge or a guardrail. Furthermore, if a particular test fails, in contraire 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 modules. This also applies to different variants of the object under test (OUT), which need individual approval with current test methods. To approve the 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. [14]. Additional 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 [15, p. 88] 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" are discretized in steps of 5 km/h within the parameter space and a "pair-wise" coverage as proposed by Schuldt et al. [16, p. 15] 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, particular tests from different scenarios can be aggregated if the test criteria/parameters are identical or subsets of the criteria/parameter from another particular test. Furthermore, the most suitable test tool (e.g. Simulation, XiL, test drive, etc.), depending on its validity, can be selected for each particular test, which also helps to reduce the approval effort.

C. Requirements for the decomposition of HAD functions

To be able to develop a functional decomposition method for HAD functions, requirements on such a method have to be defined. Wachenfeld and Winner [5, p. 433] state general requirements on a test case generation for the safety assessment of HAD functions:

1) "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 object under test (OUT) nor the behavior and properties of the environment with respect to real behavior."

2) "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."

3) "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."

4) "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 additional 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:

5) 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.

6) 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.

7) Explicit pass/fail criteria for all particular tests

When carrying out a complete 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 pass/fail criteria are prerequisites for every test case.

D. Decomposition layers and interfaces for HAD functions

In this section, the proposed functional layers for the decomposition of HAD functions and the appropriate interfaces are defined.

In this work, a decomposition of the HAD function into six layers is proposed. Decomposing the function into more layers would lead to function-specific layers and therefore result in a function-specific decomposition that is not generally applicable on various HAD functions. Another problem with a high number of decomposition layers would be to define explicit pass/fail criteria for each layer. On the other hand when using fewer layers as for example the classic three-layer decomposition "sense-plan-act", which is common in robotics [cp. 17, p. 321], the potential reduction of the approval effort will be less. The proposed decomposition is based on the five-layer decomposition by Graab et al. [13]. However, the layer information processing is split into the layers information processing and situational understanding.

1) Layer 0: Information access

This basic layer 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. Exemplary, missing or defective information from a digital map or traffic signs that are hidden by parking cars would rank among layer 0. As can be seen in the last example, the mounting positions of the environment perception sensors have an influence on their field of view, which has an influence on the information access. It can be assumed that the best achievable mounting position has already been chosen before functional testing. Nevertheless, to ensure a safe functionality of the HAD function, the information access has to be taken into account even if it is not part of the HAD function itself as failures in layer 0 have to be detected and compensated by the OUT. The interface between layer 0 and layer 1 is all the information that would be accessible for an ideal HAD system or human.

2) Layer 1: Information reception

The information reception layer contains all environment perception sensors of the OUT as well as car2x or backend

communication channels. An exemplary error to occur in layer 1 would be a dirty camera that cannot receive all accessible information. The interface between layer 1 and layer 2 are the sensor raw data or the information received via car2x or backend communication.

3) Layer 2: Information processing

Sensor fusion, object classification, and generation of an environment model are contained in the information processing layer. Typical failures in this layer would be false negative objects or object classification errors. The interface between layer 2 and layer 3 is a scene as defined by Ulbrich et al. [18].

4) Layer 3: Situational understanding

In this layer, the scene from layer 2 is extended with goal- and value-specific information selection and augmentation. Failures like wrong predicted trajectories of object vehicles can occur in layer 3. The interface to layer 4 is a situation according to [18].

5) Layer 4: Behavioral decision

This layer contains the algorithms that based on the situation model decide about the behavior of the HAD function. An exemplary failure would be an error in the maneuver planning that leads to a collision with another vehicle. The interface to the action layer is the target trajectory.

6) Layer 5: Action

The final layer transforms the trajectory from level 4 into the actual vehicle movement. It includes the vehicle's motion control algorithms as well as the necessary actuators. An exemplary failure for level 5 would be an unstable motion control algorithm.

Figure 1 gives an overview of the proposed decomposition layers and their interfaces:

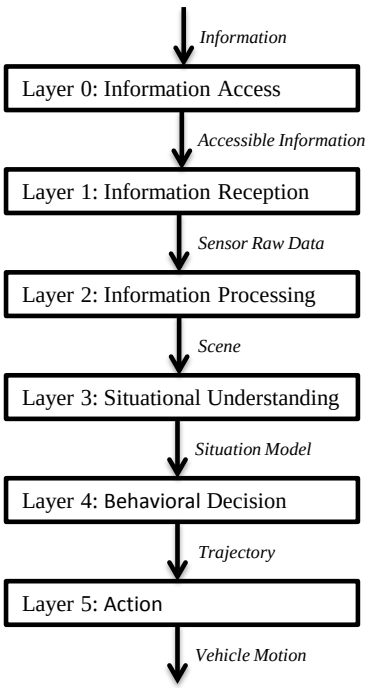

Fig. 1. Decomposition Layers

E. Defining particular test cases

To use the presented approach for the generation of test cases, following steps have to be carried out:

1) Identification of functional scenarios

As a first step, relevant scenarios to derive the test cases have to be identified. Within the project PEGASUS, relevant scenarios for the "Autobahn-Chauffeur" are identified and saved to a central scenario database [19]. Hereby it is differentiated between functional, logic, and concrete scenarios that are defined by Bagschik et al. [20] as follows:

a) Functional scenarios

"Functional scenarios describe operation scenarios […] on a semantic level. The entities and relations between the entities of the application domain are expressed in linguistic stated scenarios. […]"

b) Logic scenarios

"Logic scenarios describe operation scenarios with entities and relations between this entities based on parameter ranges in the state space.[…] Logic scenarios contain a formal description of scenarios."

c) Concrete scenarios

"Concrete scenarios describe operation scenarios explicit with entities and relations between this entities based on fix values in the state space."

2) Creation of an overview matrix

In a second step, an overview matrix is generated based on the used decomposition layers and functional scenarios for the test case generation. The decomposition layers are represented by rows and the scenarios are represented by columns. Each cell then represents a set of test cases for each allocation of scenarios and layers.

3) Definition of fail criteria

In this important step, fail criteria for each cell of the overview matrix are defined. Therefore a fault tree analysis (FTA) according to [21] with an accident as top event is conducted for each functional scenario. The base effects, which are described on a general level in this step, are then allocated to the decomposition layers and inserted in the overview matrix as fail criteria.

4) Elimination and aggregation of fail criteria

A prerequisite for this step is the conversion from functional scenarios into concrete scenarios by allocating specific parameters from the parameter space. Hereby one functional scenario can be converted to multiple concrete scenarios [20]. For each concrete scenario, the fail criteria have to be transformed into concrete fail criteria by allocating specific parameters as well. Subsequently, redundant fail criteria can be eliminated or aggregated based on an equivalence or subset analysis.

5) Selection of suitable test environments and test cases for the remaining fail criteria

For all of the remaining fail criteria, the most suitable test environment has to be selected and specific test cases have to be defined that approve the absence of the allocated fail

criteria. If it is possible to approve the nonfulfillment of all fail criteria that are deducted from all causing effects of an FTA, the absence of the top event is approved as well.

F. Exemplary application

For an exemplary application of the here presented methodology, the functional scenario that led to a real world accident of an Tesla Model S in Switzerland in May 2016 [22] is used. The scenario is visualized in figure 2:

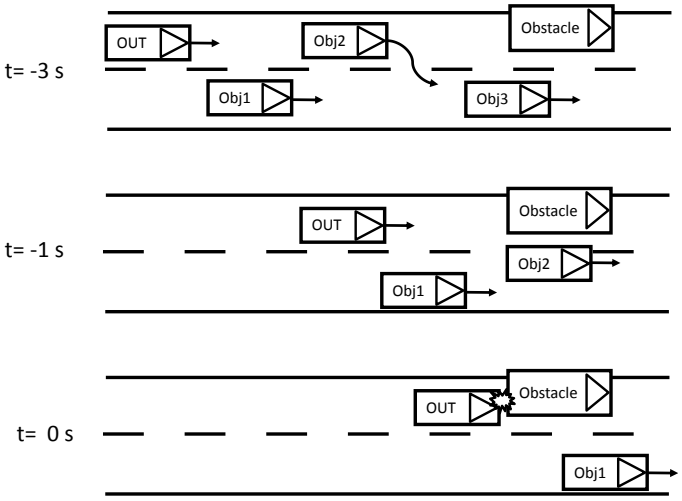

Fig. 2. Exemplary Scenario

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 vehicle in front of the OUT (Obj2) is performing a lane change to the right lane to avoid the obstacle while the OUT doesn't attempt to avoid the obstacle and is crashing into it. In this example it is assumed, that the OUT is a level 3 vehicle (according to [3]) 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 3 shows an extract from the exemplary failure tree. Starting with the collision as top event, the responsible effects 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 brake distance. This would be allocated to layer 1. As this examples shows, the base events are describe on a very general level in this step.

After collecting all base events for all relevant functional scenarios as functional fail criteria in an overview matrix (see step 2) in section II E), the functional fail criteria are transformed into concrete fail criteria by allocating specific parameters, e.g. initial velocity, friction coefficient, etc. The concrete fail criterion in this example would then be: "The sensor range is lower than *x* m". After transferring all functional fail criteria into concrete fail criteria, steps 4) and 5) as described in section II E are performed to derive particular test cases.

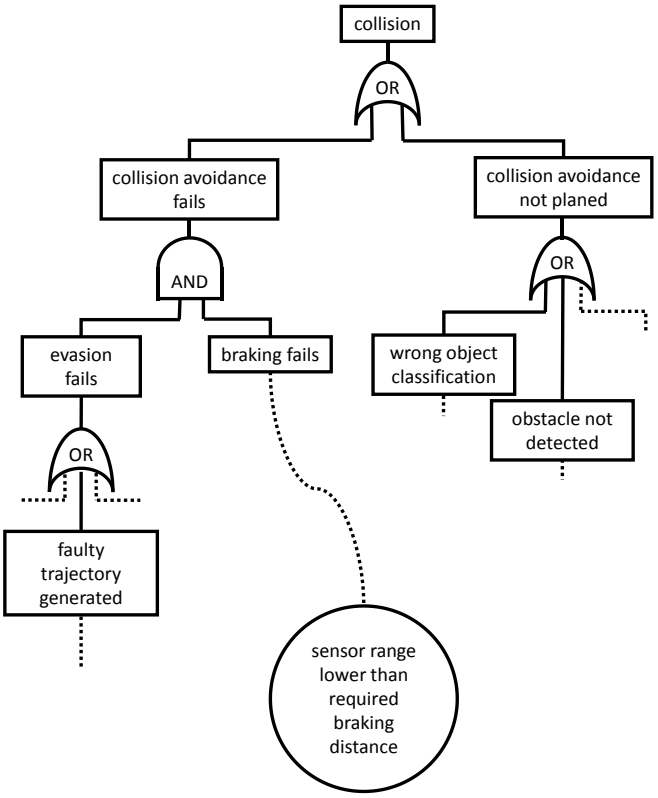

Fig. 3. Exemplary Failure Tree

III. CONCLUSION AND OUTLOOK

Test methods that are currently used to approve ADAS cannot be transferred to HAD while keeping testing effort on an acceptable level. The scenario-based approach that is developed within the research project PEGASUS promises to reduce the approval effort for HAD significantly. The decomposition approach, which is presented here for the first time, goes a step further and has the potential to reduce the approval effort even more.

By combining the scenario-based approach with a functional decomposition of the HAD function to be approved, particular test cases can be specified based on an FTA. Some of the resulting test cases can be eliminated or aggregated. Additional to this reduction of test cases, the decomposition approach can be used to reduce the approval effort for variants or updated functions.

In this paper, a generic six-layer decomposition for HAD functions is proposed based on a requirement definition. Furthermore, the potential to reduce the approval effort is outlined. Finally, a methodology to create test cases and to define corresponding fail criteria based on this decomposition and relevant scenarios is shown and applied to one exemplary scenario.

However, it still must be shown that the decomposition approach is generally applicable and not only for selected examples. This and a quantification of the potential reduction of approval effort have to be investigated in following studies.

REFERENCES

[1] J. Ziegler *et al.*, "Making Bertha Drive—An Autonomous Journey on a Historic Route," *IEEE Intell. Transport. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, 2014.

[2] J. M. Wille, F. Saust, and M. Maurer, "Stadtpilot: Driving autonomously on Braunschweig's inner ring road," in *2010 IEEE Intelligent Vehicles Symposium (IV)*, pp. 506–511.

[3] *SAE J3016: Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems*, 2014.

[4] *ISO 26262: Road vehicles – Functional safety*, 2011.

[5] W. Wachenfeld and H. Winner, "The Release of Autonomous Vehicles," in *Autonomous Driving: Technical, Legal and Social Aspects*, H. Winner, M. Maurer, J. C. Gerdes, and B. Lenz, Eds., Berlin, Heidelberg: Springer, 2016, pp. 425–449.

[6] A. Reschka, G. Bagschik, S. Ulbrich, M. Nolte, and M. Maurer, "Ability and skill graphs for system modeling, online monitoring, and decision support for vehicle guidance systems," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*, 2015, pp. 933–939.

[7] T. M. Gasser, A. Seeck, and B. W. Smith, "Framework Conditions for the Development of Driver Assistance Systems," in *Handbook of Driver Assistance Systems*, H. Winner, S. Hakuli, F. Lotz, and C. Singer, Eds., Cham: Springer International Publishing, 2016, pp. 35–68.

[8] M. Christiansen, *In geheimer Mission: auf Abnahmefahrt mit der neuen Mercedes E-Klasse, W213.* [Online] Available: http://5komma6.mercedes-benz-passion.com/in-geheimer-mission-auf-abnahmefahrt-mit-der-neuen-mercedes-e-klasse-w213/. Accessed on: Jul. 12 2017.

[9] N. Kalra and S. M. Paddock, "Driving to Safety: How Many Miles of Driving Would It Take to Demonstrate Autonomous Vehicle Reliability?," 2016. [Online] Available: http://www.rand.org/pubs/research_reports/RR1478.html.

[10] German Aerospace Center (DLR), *"PEGASUS RESEARCH PROJECT"*. [Online] Available: http://pegasus-projekt.info/en/home. Accessed on: Jun. 11 2017.

[11] P. Junietz, J. Schneider, and H. Winner, "Metrik zur Bewertung der Kritikalität von Verkehrssituationen und - szenarien," in *Workshop Fahrerassistenz und automatisiertes Fahren*, 2017.

[12] F. G. O. Lotz, "Eine Referenzarchitektur für die assistierte und automatisierte Fahrzeugführung mit Fahrereinbindung," Dissertation, Technische Universität Darmstadt, 2017.

[13] B. Graab, E. Donner, U. Chiellino, and M. Hoppe, "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.

[14] U. Baake, K. Wüst, M. Maurer, and A. Lutz, "Versuchs-und simulationsbasierte Absicherung von ESP-Systemen für Transporter," *ATZ-Automobiltechnische Zeitschrift*, vol. 116, no. 2, pp. 46–51, 2014.

[15] Y. Lu, "Trajektorienplanung und Fehlerursachenanalyse für die automatisierte Autobahnfahrt in der Simulation," Masterthesis, TU Darmstadt, Fachgebiet Fahrzeugtechnik, 2015.

[16] F. Schuldt, F. Saust, B. Lichte, M. Maurer, and S. Scholz, "Effiziente systematische Testgenerierung für Fahrerassistenzsysteme in virtuellen Umgebungen," in *Automatisierungssysteme, Assistenzsysteme und eingebettete Systeme für Transportmittel (AAET)*, Braunschweig, 2013.

[17] J. Hertzberg, K. Lingemann, and A. Nüchter, *Mobile Roboter*: Springer Berlin Heidelberg, 2012.

[18] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, "Defining and substantiating the terms scene, situation, and scenario for automated driving," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, 2015, pp. 982–988.

[19] A. Pütz, A. Zlocki, J. Bock, and L. Eckstein, "System validation of highly automated vehicles with a database of relevant traffic scenarios," in *12th ITS European Congress*, 2017.

[20] G. Bagschik, T. Menzel, A. Reschka, and M. Maurer, "Szenarien für Entwicklung, Absicherung und Test von automatisierten Fahrzeugen," in *Workshop Fahrerassistenz und automatisiertes Fahren*, 2017.

[21] *IEC 61025, Fault Tree Analysis (FTA)*, 2006.

[22] F. Lambert, *Tesla Model S driver crashes into a van while on Autopilot [Video]*. [Online] Available: https://electrek.co/2016/05/26/tesla-model-s-crash-autopilot-video/. Accessed on: Aug. 31 2017.