Multi-aspect Safety Engineering for Highly Automated Driving

Looking Beyond Functional Safety and Established Standards and Methodologies

Patrik Feth[1(✉)], Rasmus Adler[1], Takeshi Fukuda[2], Tasuku Ishigooka[2], Satoshi Otsuka[2], Daniel Schneider[1], Denis Uecker[1], and Kentaro Yoshimura[2]

[1] Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern, Germany
patrik.feth@iese.fraunhofer.de
[2] Hitachi, Ltd., Tokyo, Japan

Abstract. Highly automated and autonomous driving is a major trend and vast amounts of effort and resources are presently being invested in the development of corresponding solutions. However, safety assurance is a concern, as established safety engineering standards and methodologies are not sufficient in this context. In this paper, we elaborate the fundamental safety engineering steps that are necessary to create safe vehicles of higher automation levels. Furthermore, we map these steps to the guidance presently available in existing (e.g., ISO26262) and upcoming (e.g., ISO PAS 21448) standards and point out open gaps. We then outline an approach for overcoming the identified deficiencies by integrating three different safety engineering disciplines. This includes (1) creating a safe nominal behavior specification; (2) dealing with functional insufficiencies, and (3) assuring the related performance wrt. functional safety. We exemplify our proposed methodology with a case study from industry.

1 Introduction

In many embedded systems domains we presently see a trend towards higher levels of automation up to the point of autonomy. Highly automated and autonomous driving, for instance, is a major trend, and vast amounts of effort and resources are presently being invested in the development of corresponding solutions. Significant progress has already been made and results have been very promising; for instance, numerous demonstrator vehicles have impressively shown the technical feasibility of advanced automated driving features up to the point of fully autonomous driving. However, before such features can become actual products, it is absolutely mandatory to ensure that they do not introduce unacceptable levels of risk. This is the domain of safety engineering, where we presently face major challenges due to the insufficiency of established methods and standards, which are mostly designed with respect to aspects of functional safety, omitting other aspects that are now gradually moving into the limelight.

B. Gallina et al. (Eds.): SAFECOMP 2018, LNCS 11093, pp. 59–72, 2018.
https://doi.org/10.1007/978-3-319-99130-6_5

Considering only functional safety is not enough because systems are no longer fully controlled by human operators; rather, they increasingly incorporate their own extended perception, reasoning, and decision capabilities. In the automotive domain, this trend manifests in the introduction of vehicles with increasing levels of automation. According to the SAE classification [9], the currently available Advanced Driver Assistance Systems (ADAS) can be classified as level 2 or partially automated systems. However, Audi is the first manufacturer [2] claiming to be technically ready for automation level 3 as soon as the corresponding regulations are available. The transition from automation level 2 to automation level 3 is a remarkable change, especially from a safety point of view. This is due to the fact that starting from level 3, the system is actually responsible for rendering safe nominal behavior. In contrast to that, for lower automation levels, it is still the driver who is responsible for guaranteeing safe system behavior and the safety scope is consequently limited to functional safety, i.e., to hazards caused by (random and systematic) faults. This change in responsibility changes the way we need to perform the overall engineering and especially the safety engineering of those systems.

Thus, for lower automation levels, the driver is responsible for choosing an appropriate vehicle behavior in a given driving situation and the vehicle is responsible for supporting the driver in terms of situation awareness and correct implementation of the driving decisions. Any electric or (programmable) electronic system (E/E system) that affects controllability by the driver is consequently considered safety-critical. This obviously includes also ADAS, as the goal of any ADAS is to assist the driver and contribute to his ability to control the vehicle. The topic of safety assurance of such E/E systems is well studied and corresponding guidance is provided by the existing safety standard ISO 26262 and the upcoming safety standard ISO PAS 21448 “Road vehicles - Safety of the intended functionality” (SOTIF). ISO 26262 focuses on functional safety, which means managing risks emerging from malfunctioning behavior (due to random hardware failures or systematic failures) of E/E systems. It does not, however, cover safety issues emerging from functional insufficiencies. This means that it assumes that the performance limits of all functions are specified in a reasonable and safe manner, so that it is sufficient to focus on critical deviations of the specified functionality. Particularly for ADAS, with their complex situation awareness, this assumption becomes very difficult to handle and hence leads to a gap in the established field of safety assurance. The upcoming SOTIF standard is meant to close this gap. For example, a camera without a night filter can only work during daytime or a LIDAR sensor might not work in heavy snowfall or even in rain. Thus, any creation of situation awareness based on these sensors will fail in these respective critical situations. SOTIF addresses this problem by supporting the systematic selection of an appropriate sensor concept.

However, the basis for conceiving an adequate sensor concept and related performance limits is knowing which situations need to be detected with which level of confidence. Furthermore, it is necessary to define appropriate responses if the current situation cannot be determined/classified at all or not with sufficient

confidence. In the case of automation levels 1 and 2 and partly also in the case of level 3, the response can be a shutdown of the functionality and transition to manual driving. This strategy is not applicable at automation level 4, as any fallback to manual driving is excluded by definition.

However, even if we assume that the correct detection of situations is not a problem, we still have to deal with a huge number of driving situations if we want to define which vehicle behavior is appropriate in which situation. We use the term **safe nominal behavior specification** to refer to such a specification that describes which driving behavior is safe in which situation and that abstracts from all technological challenges of situation awareness. Existing standards, including the upcoming SOTIF, provide no guidance for engineering such a safe specification even though the name SOTIF seems to indicate at least some support in this regard.

We think that being aware of these different dimensions within the overall notion of safety is very important to foster a structured discussion and to organize further research in these important fields. Furthermore, it helps to avoid over- or misinterpretations of existing safety standards in the context of safety engineering for higher automation levels.

The remaining article is structured as follows: Sect. 2 highlights the contribution of this paper and places it in the context of related work. Section 3 gives a brief overview of the solution and points out some of its risks and limitations. Section 4 presents the proposed solution and a holistic approach that achieves the above-mentioned goals. Section 5 relates this process to SOTIF and other safety standards. Section 6 exemplifies the proposed solution before Sect. 7 concludes the paper.

2 Related Work

To the best of the authors knowledge, the state of the art is still lacking wrt. precise identification and characterization of the gaps in current safety methods and standardization regarding highly automated and autonomous driving (and systems in general). Existing work in this field rather provides experience reports on the usage of ISO 26262 for vehicles of higher automation levels, for example in [10] Spanfelner et al. identify insufficient models as the major problem of using ISO 26262 for driver assistance systems. We agree with this line of argumentation and add a contribution to their work, as we additionally consider the ISO PAS 21448 “Safety of the Intended Functionality” standard and propose a holistic process that goes beyond ISO 26262 instead of enforcing the use of existing standards (which clearly have not been designed for highly automated or autonomous systems). Higher-level thoughts on the topic of safety for vehicles of higher automation levels can be found, for example, in [4]. In this work, Koopman and Wagner also mention the shortcomings of ISO 26262, but do not provide clear concepts or methods on how to overcome these problems.

In contrast to this, one major aspect of our work in this paper is the identification of the need to specify safe system behavior, i.e., to create a safe nominal

behavior specification. Although we propose the usage of state machines to this end, we do not intend to argue that this is the best way to do it. We argue that the non-existence of dedicated methods and standards for this aspect indicates that there is currently no commonly accepted best practice, and it will be the task of future experience to find such a practice. In earlier work, Leveson also used state machines to describe the higher-level behavior of a safety-critical system [7]. In more recent work, Leveson uses control structure diagrams in the Systems-Theoretic Accident Model and Processes (STAMP) [5] and the related Systems-Theoretic Process Analysis (STPA) [6] approach for hazard identification. These approaches build on a systems engineering foundation for analysis and, just like ISO 26262 and the SOTIF standard, focus on deviations from this specification of the intended behavior, which is assumed to be safe. Some earlier work developed at Fraunhofer IESE is systematizing and automating hazard and risk analysis [3]. This might also help in analyzing highly automated or autonomous systems, where degrees of freedom and uncertainties lead to a very high complexity, which is generally hard to tackle without systematic, tool-supported and ideally (semi-) automated approaches.

3 Safety Aspects Relevant to Autonomous Systems

As illustrated in Fig. 1, we argue that guidance (by means of methods, techniques and maybe explicit standardization) is required for the creation of a safe behavior. The aspect of creating a safe nominal behavior specification has been out of scope for existing safety standards and ongoing activities in standard creation initiatives (e.g., SOTIF), but is becoming very important for highly automated and autonomous systems. The safe nominal behavior specification defines which behavior is safe in which situation and is therefore the basis for reasoning about functional performance limits and which limits are sufficient and which are insufficient.

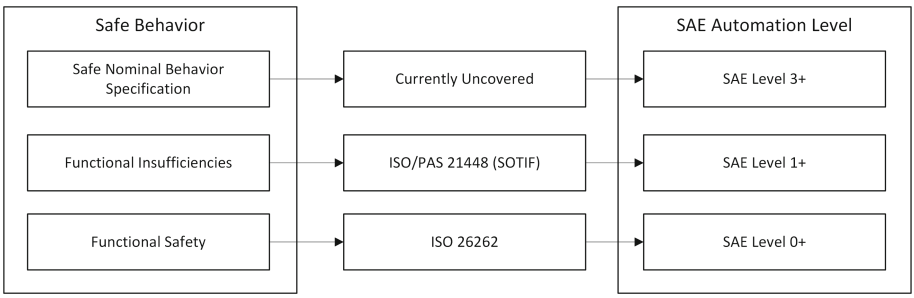

Fig. 1. Elements to reach safe behavior: standard coverage

The second layer provides guidance for dealing with functional insufficiencies, which is likewise the focus of the upcoming SOTIF standard. The performance

limits in the SOTIF wording are of a more technical nature than the functional performance limits that we consider as part of the safe nominal behavior specification. Both functional and technical performance limits are a prerequisite to reasoning about functional safety and considering violations with respect to these performance limits.

The third important safety aspect for autonomous systems is functional safety, which is the predominant aspect considered in the engineering of todays (more or less) operator controlled systems. Consequently, there is already a broad established and proven range of techniques, methods, standards and tools available which can be used (or be at least a starting point) also for autonomous systems. Of course, all three relevant safety aspects should be integrated and harmonized on a conceptual and methodological level and corresponding standardization shall be tightly interlinked as it is, for instance, already the case for the ISO 26262 and the ISO PAS 21448.

The three safety aspects elaborated above can be further illustrated in mapping them to the standard architecture of embedded systems (Monitor - Plan - Actuate). Doing this leads to the following observation: The SOTIF safety aspect (and thus the ISO PAS 21448) focuses on the monitoring part and provides guidance for creating a safe situation awareness. The functional safety aspect (and thus the functional safety standard ISO 26262) focuses on random and systematic software and hardware failures that might impact any element of the cycle, i.e. monitoring, planning, and actuation. A significant gap exists regarding the assurance of safety with respect to automated or autonomous planning. The planning determines which vehicle maneuvers and trajectories are safe in which (perceived) situations. It is thus closely related to what we call the aspect of safe nominal behavior specification in this article. The planning needs to implement the safe specification as well as possible, particularly considering the uncertainties that are dynamically induced by the monitoring element due to inherent challenges in assessing the current context situation.

In summary, we argue that the engineering of a safe highly automated or autonomous system requires the consideration of functional safety, functional insufficiencies, and a safe nominal behavior specification. Today, only the first aspect of functional safety is well understood, supported by established methods, techniques and tools and addressed by an existing and established standard: the ISO 26262. So it is known only for this aspect what is considered necessary to claim sufficient coverage, i.e. to be able to argue a sufficient level of functional safety to release a (non-automated) car as a product. For the topic of functional insufficiencies, a draft version of an upcoming standard, ISO PAS 21448 “Road vehicles - safety of the intended functionality”, was used as input for deriving the recommendations in this article. The requirements concerning this topic might change in the future; best practices are not known yet and need to be established over time. The current draft version of the SOTIF standard is explicitly only intended to cover vehicles up to automation level 2. On top of the aspects considered by SOTIF and ISO 26262, the aspect of how to create a safe nominal behavior specification needs to be addressed for higher automation levels.

At the time of this writing, this aspect is not being considered yet at all by existing standards or by standard creation initiatives. One reason for that might be, that in non-automated system a safe nominal behavior is typically pretty straight forward and commonly agreed upon. In case of a car, it is clear how the user interface looks like and how a driver operates it. And it is also clear, that monitoring and planning are tasks of the driver and the driver is thus responsible for the driving behavior, leaving only functional safety as important safety aspect. Now, given the trend towards ever higher levels of automation across domains, this area requires more consideration in the future to allow the development and also validation (i.e. creation of sufficient evidence wrt. safety) of safe autonomous vehicles.

4 Multi-aspect Safety Engineering for Autonomous Systems

The proposed approach still generally aligns with the established principles of how safety engineering works and interacts with "normal" system engineering. Safety engineering builds upon the initial results from system engineering and analyzes them with respect to safety. Based on the analysis results, safety requirements are elicited, a safety concept is compiled, corresponding safety measures are selected, implemented and validated and a related safety argumentation is created. This procedure (or at least parts thereof) occurs at different development stages (or abstraction levels) in parallel (and interaction) with the system engineering activities. However, compared to the established approach focused on functional safety, the boundaries between safety engineering and "normal" system engineering, i.e. the engineering of the nominal system behavior, are softened. In particular with respect to the engineering of a safe nominal behavior (of the automated behavior) we obviously have a tighter integration between the disciplines.

As illustrated in Fig. 2, we consider three (horizontal) abstraction layers that directly relate to the three safety aspects identified in the previous section.

The System in Its Usage Context layer is related to the aspect of defining a safe nominal behavior specification. It represents an abstraction layer on which high-level concepts of the system and the requirements on the system are described independent of their technical realization.

The System Realization Concept layer is related to the aspect of handling functional insufficiencies. It contains first technical information on the future system, such as sensor concept and algorithms.

Finally, the System Functional View layer is related to the aspect that addresses functional safety. It represents the more detailed functional view on the system and describes how the requirements are functionally realized by the system. It is the basis for conducting safety analyses and deriving a functional safety concept in terms of ISO 26262.

The first fundamental research question concerns the notations and modeling languages used for representing the system at the different abstraction layers.

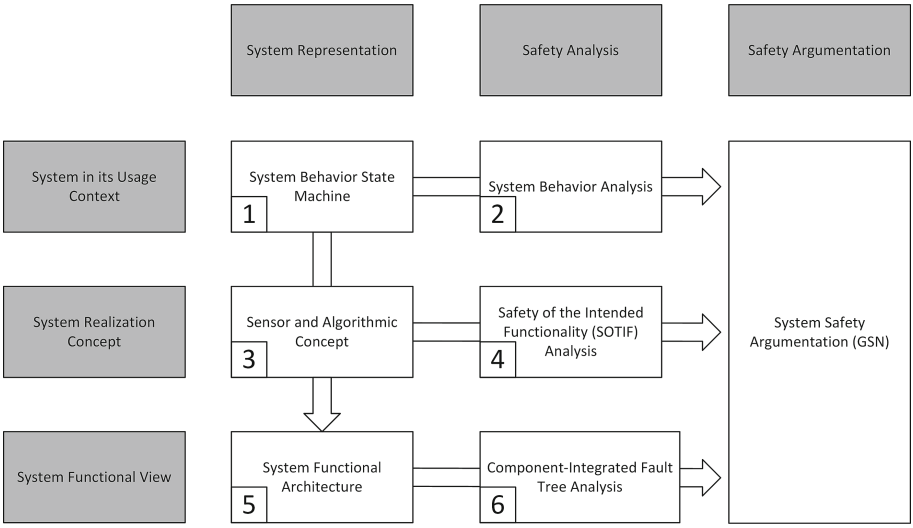

Fig. 2. Multi-aspect safety engineering process

At the highest level of abstraction, we aim at a definition of the vehicle motion. In a variety of industry projects, we have observed that this is often specified via images showing trajectories of the ego-vehicle in a certain driving situation. The problem with this approach is the assurance of completeness. All possible sequences of valid trajectories need to be specified. To solve this completeness problem, we propose state machines, as state machines are a common notation for specifying complete sets of sequences. Furthermore, this specification technique supports modeling the activation, deactivation, and degradation of the automation functions. At the more detailed abstraction level, we recommend using the familiar notation of functional architectures. However, creating these system representations is not within the actual scope of safety engineering and should be performed, or at least strongly assisted, by domain and system engineering experts.

The safety analysis is conducted on the basis of the system representation. At the requirements level, this is done by analyzing the behavior of the system and deriving possible hazards of that behavior independent of the technical realization. The next layer, the System Realization Concept layer, already contains initial information on the technical realization and considers this information in the safety analysis. The SOTIF analysis additionally and explicitly takes into account functional insufficiencies of the technology used. More details on how to perform the safety analysis in these two top layers will be given with the example in Sect. 6. At the functional level, we propose using component-integrated fault trees (CFT) to refine high-level safety goals into more detailed functional safety requirements. CFTs have proven their benefit in multiple industry projects and are an accepted approach for systematic and model-driven safety analysis [1].

The Safety Argumentation layer is the bonding element among the three abstraction layers. The safety argumentation can also be considered as a safety documentation as it stores the refinement of top-level safety goals into safety requirements and provides a structured argument regarding how the fulfillment of low-level requirements implies fulfillment of top-level goals. As a notation, we recommend the Goal-Structuring Notation (GSN) for this activity [8].

The order for the outlined activities in the previous section depends on their dependencies. Activities at higher levels of abstraction depend on activities at lower levels of abstraction and vice versa. Furthermore, safety analyses depend on the engineering of the system representation being analyzed, and the selection of safety measures along with the related safety argumentation depends on the safety analyses. Any process that take these dependencies into account is valid. We assume the following waterfall-like process:

1. Model high-level system concept
   - Output: State machine model of the behavior of the system independent of any implementation details
2. Consider the high-level system behavior and possible hazards of this behavior
   - Output: Safe system specification including safety goals derived by a systematic state space analysis
3. Provide information on the sensor and algorithmic concept
   - Output: Sensor and algorithmic concept as a basis for SOTIF analysis
4. Conduct analysis of system limitations and functional insufficiencies
   - Output: Safe system specification extended with consideration of system limitations and functional insufficiencies and their derived safety goals
5. Model functional architecture consistent with safe system specification and realization concepts
   - Output: System functional view as a basis for ISO 26262 analysis
6. Perform ISO 26262 analysis to investigate the contribution of component failures to the violation of safety goals
   - Output: Functional safety requirements assigned to components in the architecture.

Steps 1 and 2 take place at the highest abstraction levels and are independent of any implementation details. Steps 3 and 4 take place at the SOTIF level and consider realization concepts. Steps 5 and 6 take place at the ISO 26262 functional safety level.

5 Relation to SOTIF and Other Standards

In the following, we relate the six steps of the solution proposed in the previous section to the current world of safety standardization. To this end, we look at the scope definition in the current draft of the SOTIF standard. This scope definition considers different causal factors of hazards, providing the relation to SOTIF and other standards for each factor. The casual factors are as follows:

1. E/E system failures
2. Unintended behavior without fault or failure (including E/E system performance limit)
3. Foreseeable user misuses
4. Security violation
5. Impact from active infrastructure and/or vehicle to vehicle communication
6. Impact from car surroundings
7. Unsafe nominal behavior specification.

We enhanced this overview with respect to the safe nominal behavior specification and structured it using the Goal Structuring Notation (GSN). Due to space limitations, we cannot show the full GSN here. From the top-level goal "Perform such safety engineering activities that guarantee the absence of unreasonable risk for the automation level 4 driving system", we derived "the safe specification" by creating subgoals related to the aforementioned causal factors in the SOTIF scope and the scope of this work. The defined safety goals were then defined as "Perform such safety engineering activities that guarantee the absence of unreasonable risk for the automation level 4 driving system caused by [element from the enumeration before]".

The second level of refinement argues over existing safety standards for the particular source of unreasonable risk. For security violations (safety goal 5), the "Cybersecurity Guidebook for Cyber-Physical Vehicle Systems" (J3061) has been an initial step (which is presently being integrated in other activities), and for vehicle-to-infrastructure and vehicle-to-vehicle communication (safety goal 6), the "Road Vehicle - Extended Vehicle" ISO 20077 standard is about to be published. Performing safety engineering activities according to the recommendations in these standards leads to an absence of unreasonable risk caused by the particular aspect addressed by this standard. In the project we conducted, it was assumed that the vehicle is not connected to its environment. Because of that, both causes were out of scope in the conducted project and there was no further refinement of the goals to perform safety engineering activities according to the safety standard.

The cause of E/E system failures (safety goal 1) is addressed by the ISO 26262 standard. The SOTIF standard claims to address the causes of unintended behavior without fault or failure, foreseeable user misuse, and impact from car surroundings (safety goals 2–4).

The final level of refinement contains the steps of the proposed solution in Sect. 4. The performance of activities belonging to the concept phase from ISO 26262 and the SOTIF standard is given as the goals at this refinement. The activities of the standard are mapped to the activities in the solution. Note that not every step in the suggested process can be mapped to an existing standard. This is due to the fact that no standards are currently available for the development of systems with higher automation levels. Even the SOTIF standard is only suited up to automation level 2 to date. Considering this background, the suggested process might need to be revised once appropriate standards have been established.

6 Methodology Example

This section exemplifies the solution presented in Sect. 4 for a highway assistant system. The system under consideration is classified as SAE automation level 4 and shall operate without interruption and without relying on a human driver as fallback performance of the dynamic driving task but with the limited system capability to operate on highways only. This system will be used as an ongoing example in this section.

6.1 Model High-Level System Concept

The first step in the proposed process is the modeling of the high-level system concept. The goal of this step is to capture the system concept at a high abstraction level. This includes the concept for activation, deactivation, degradation (e.g., from level 4 to level 3), and for handling emergency situations. In addition, it includes the general vehicle behavior in these different cases. As mentioned above, we propose using state charts to represent these high-level concepts. In the conducted project, we created such a state chart for an automation level 4 highway assistant system.

6.2 Consider the High-Level System Behavior and Possible Hazards of This Behavior

After the system's behavior has been captured at a high abstraction level, it can be analyzed regarding its safety in different driving situations. Whether a behavior is safe or appropriate depends on the current situation. For example, the operating mode "passing" is a type of behavior that is not safe or unsafe independent of the situation. It is a safe and appropriate behavior if there is a slower vehicle in front and the left lane is free; but it is an unsafe behavior if a vehicle is currently approaching on the left lane. This analysis of the safety of behavior in different situations has to be conducted for systems of higher automation levels. For each operating mode, i.e., for each state in the above state chart, one needs to argue on the preconditions that must be fulfilled to enter a mode and the circumstances under which a mode needs to be deactivated. **A mode shall only be activated if the risk of this mode in the current driving situation is acceptable; if the risk of the mode becomes unacceptable, the mode needs to be deactivated**. Deactivation obviously requires us to define a mode that is less risky. If there is no other alternative how the vehicle can drive "safer" in the considered situation and if this "safest" solution is not acceptable, then we have to think about external measures that can serve to avoid the occurrence of the situation (e.g., external infrastructure, new driving laws, etc.). As the introduction of external measures goes beyond the scope of this report, we focus on patterns for annotating the operating modes with safety conditions and assumptions.

This general line of thought directly gives us a pattern for deriving a safe nominal behavior specification. To make this step systematic, we recommend performing a systematic analysis of the state space that the system can encounter. A possible way to do this is the usage of tables describing environmental factors and possible values for these factors. For each value or combination of these values, a classification is performed as to whether it is acceptable to allow the operating mode or whether the situation requires deactivation of this mode.

For the operating mode "Passing" of the AL 4 driving system, an analysis of the behavior in different situations has been conducted. An excerpt of this analysis shows that it may yield the following safety goals for passing:

- Passing must not be performed at an intersection (merge) area of a highway
- Passing must not be performed if there is a vehicle on the adjacent left lane.

6.3 Provide Information on the Sensor and Algorithmic Concept

Up to this point, the process steps have abstracted from the implementation concepts. To conduct the SOTIF analysis, these concepts need to be added to the information available about the developed system. In particular, the standard focuses on sensors and algorithms for the creation of situation awareness. Concepts about this part of the system are necessary to perform an analysis on the limitations of situation awareness. In its current version, the SOTIF standard mainly focuses on functional insufficiencies: situations in which sensors and algorithms are operating outside their intended state space. It needs to be specified how the sensors are used to create the needed situation awareness. Which situations the system needs to be aware of from a safety perspective can be derived from the analysis conducted in the step before. From the safety goals derived above for the operating mode "Passing", the following requirements on situation awareness can be derived:

- Detect intersection (merge) area of a highway
- Detect vehicle on the adjacent left lane.

The resulting sensor concept for the automation level 4 system might state that the intersection (merge) area of a highway shall be detected with GNSS and 3D maps and the presence of vehicles on the adjacent left lane shall be detected with radar, lidar and camera.

6.4 Analysis of Limitations and Functional Insufficiencies

Based on the sensing concept, the performance limits are derived. This shall be done for each sensor used. Reaching the performance limits of a sensor can again trigger a transition in the system's functional concept. An example is the usage of lidar in situations where there is heavy snowfall. Under such conditions, a lidar sensor usually does not work anymore. If the lidar sensor is the only way to determine the distance to objects in front of the vehicle, then this automation

level 4 functionality cannot be provided without this sensor. Thus, the environmental situation of heavy snowfall demands the deactivation of the AL 4 driving mode. This step refines the step of creating a safe system specification by adding implementation-specific information to the system specification.

Above, we derived a sensor concept from the safety goals related to the operating mode of "Passing". As part of the system limitations and functional insufficiencies analysis, we will detail this sensor concept. Let us assume that the sensors that are used come with the following limitations:

- Camera: Limited performance during nighttime
- Lidar: Limited performance in heavy rain and snow
- Radar: Limited performance in heavy snow
- 3D Maps: Information is usually delayed by at least 10 min
- GNSS: Limited performance inside tunnels.

Table 1 gives resulting limitations from the sensor concept.

Table 1. Sensor concept limitations table

Situation	Sensor concept	Resulting limitation
Intersection (merge) area of a highway	GNSS + 3D maps	Not possible to detect if currently at an intersection (merge) if currently driving in a tunnel due to missing GNSS reception
Vehicle on the adjacent left lane	Radar + lidar + camera	Not possible to detect vehicle on the adjacent left lane during nighttime with heavy snow due to sensor limitations

From the resulting limitations, we can derive the following functional improvements:

- Passing must not be performed while driving in a tunnel (not able to detect if currently at intersection (merge) area of a highway)
- Passing must not be performed during nighttime with heavy snow (not able to detect vehicle on the adjacent left lane or tail vehicle at traffic jam or obstacles on the road).

These safety goals become part of the safe nominal behavior specification.

6.5 Model Functional Architecture Consistent with Safe Nominal Behavior Specification and Realization Concepts

After the system behavior has been defined in a safe nominal behavior specification containing both implementation-independent information from steps 1 and 2 of the suggested solution in Sect. 4 and implementation-specific information from steps 3 and 4, this specification shall be translated into a functional

architecture as a basis for ISO 26262 analysis. Again, we do not see this step as a genuine safety engineering step but as a step to be conducted as part of the engineering process. The functional architecture shall use hierarchy and make intensive use of ports. In industry, components are often modeled only with one input- and one output-port. This is not enough to support component-integrated fault tree analysis. The information that is exchanged between the functions in the functional architecture needs to be defined in more detail. For every information with a unique character, a special port has to be created.

6.6 Perform ISO 26262 Analysis to Investigate the Contribution of Component Failures to the Violation of Safety Goals

In order to achieve the requirements of functional safety, which is of course still important for systems with a high automation level, the ISO 26262 standard is the corresponding reference in the automotive domain. This step is already standardized and mature methodologies exist to support it. We argue that the problems encountered when applying the standard to higher automation levels, which are mentioned in other publications, originate mainly from the imprecise definition of the intended function. If the steps recommended in this work are followed and a functional architecture is created that realizes a safe nominal behavior specification, ISO 26262 can be applied.

7 Conclusion

In the automotive domain, as well as in other domains of embedded systems, we see a significant trend towards ever higher levels of automation up to the point of autonomy. The economical and societal potential is huge, but several challenges need to be tackled before such systems can actually become products and a business success. One important challenge is ensuring safety, whereas established methods and standards have been designed with manually controlled systems in mind and need to be augmented to actually cover all relevant aspects for highly automated and autonomous systems.

Accordingly, in practice, safety engineering is currently mainly concerned with ensuring functional safety and the corresponding fulfillment of normative requirements from standards and regulations. Regarding systems with high automation levels, this limitation of safety engineering is not appropriate anymore. In this paper, we propose a multi-aspect safety engineering approach for highly automated driving which incorporates additional relevant safety aspects beyond functional safety and thus beyond established methods and standardization. Most importantly, we introduce the additional aspect of engineering a safe nominal behavior specification with the help of state machines and a systematic state space analysis. This puts an additional layer on top of the safety aspects tackled by ISO PAS 21448 and ISO 26262, i.e. safety of the intended functionality (actually focused on functional insufficiencies and assuming the availability of a specification of safe nominal behavior as a starting point) and functional

safety. The overall approach has been illustrated based on an industrial case study of an advanced driver assistance system. I.e. we briefly described how the safe nominal behavior specification was created, how it has been used as a starting point for the analysis of causes and consequences of deviations from the intended functionality as per the SOTIF standard and, finally, how functional safety can be tackled.

We see the core contribution of this paper in the discussion of the necessary safety considerations for highly automated systems and the explicit identification of the three required safety aspects. In doing so, we point out the current gaps in the established safety engineering state of the practice and standardization. The proposed solution is in its details still relatively premature and has only been applied in few occasions. However, the experiences made have been promising and we think that the described approach can contribute as a basis for discussion and be a starting point for further work to facilitate systematic engineering of safe highly automated and autonomous systems.

References

1. Adler, R., Schneider, S., Hoefig, K.: Evolution of fault trees from hardware safety analysis to integrated analysis of software-intensive control systems. In: International Conference on Engineering Sciences and Technologies (2004)
2. Audi (2017). https://www.audi-mediacenter.com/en/press-releases/the-new-audi-a8-future-of-the-luxury-class-9124
3. Kemmann, S.: SAHARA: a structured approach for hazard analysis and risk assessments. Dissertation. TU Kaiserslautern, Kaiserslautern (2015)
4. Koopman, P., Wagner, M.: Autonomous vehicle safety: an interdisciplinary challenge. IEEE Intell. Transp. Syst. Mag. **9**(1), 90–96 (2017)
5. Leveson, N.: A new accident model for engineering safer systems. Saf. Sci. **42**(4), 237–270 (2004)
6. Leveson, N.G.: An STPA primer. http://sunnyday.mit.edu/STPA-Primer-v0.pdf
7. Leveson, N.G., Heimdahl, M.P., Hildreth, H., Reese, J.D.: Requirements specification for process-control systems. IEEE Trans. Softw. Eng. **20**, 684–707 (1994)
8. Limited, O.C.Y.: GSN community standard version 1 (2011)
9. SAE: J3016: Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles (2016)
10. Spanfelner, B., Richter, D., Ebel, S., Wilhelm, U., Branz, W., Patz, C.: Challenges in applying the ISO 26262 for driver assistance systems. Schwerpunkt Vernetzung, 5. Tagung Fahrerassistenz (2012)